

DB2 Universal Database for OS/390 and z/OS



# Installation Guide

*Version 7*



DB2 Universal Database for OS/390 and z/OS



# Installation Guide

*Version 7*

**Note**

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 541.

**Second Edition, Softcopy Only (August 2001)**

This edition applies to Version 7 of IBM DATABASE 2 Universal Database Server for OS/390 and z/OS (DB2 for OS/390 and z/OS), 5675-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.

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 book since the hardcopy book was published are indicated by the hash (#) symbol in the left-hand margin. Editorial changes that have no technical significance are not noted.

This and other books in the DB2 for OS/390 and z/OS library are periodically updated with technical changes. These updates are made available to licensees of the product on CD-ROM and on the Web (currently at [www.ibm.com/software/data/db2/os390/library.html](http://www.ibm.com/software/data/db2/os390/library.html)). Check these resources to ensure that you are using the most current information.

© Copyright International Business Machines Corporation 1982, 2001. All rights reserved.

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

---

# Contents

<b>About this book</b> . . . . .	xv
Who should read this book . . . . .	xv
How to use the DB2 library . . . . .	xv
How to obtain DB2 information . . . . .	xvii
DB2 on the Web . . . . .	xvii
DB2 publications . . . . .	xviii
DB2 education . . . . .	xviii
How to order the DB2 library . . . . .	xviii
Subscription through the Publication Notification System (PNS) . . . . .	xix
Product terminology and citations . . . . .	xix
How to read the syntax diagrams . . . . .	xx
How to send your comments . . . . .	xxi
 <b>Summary of changes to this book</b> . . . . .	 xxiii

---

## Part 1. Introduction . . . . . 1

<b>Chapter 1. Summary of changes to DB2 for OS/390 and z/OS Version 7.</b> . . . .	3
Enhancements for managing data . . . . .	3
Enhancements for reliability, scalability, and availability . . . . .	3
Easier development and integration of e-business applications . . . . .	4
Improved connectivity . . . . .	5
Features of DB2 for OS/390 and z/OS . . . . .	6
Migration considerations . . . . .	6
 <b>Chapter 2. Introduction to installation and migration</b> . . . . .	 7
Installing other features of the DB2 UDB Server for OS/390 and z/OS . . . . .	7
Installation features . . . . .	7
Installation and migration steps overview . . . . .	8
Planning for migration incompatibilities . . . . .	9
SMP/E steps summary . . . . .	9
Installation steps summary . . . . .	10
Migration steps summary . . . . .	10
Fallback steps summary . . . . .	12
Remigration steps summary . . . . .	12

---

## Part 2. Planning and installing DB2 . . . . . 15

<b>Chapter 3. Estimating DB2 storage needs</b> . . . . .	25
Disk storage for the DB2 subsystem . . . . .	25
Disk requirements for the DB2 catalog . . . . .	26
Disk requirements for the DB2 directory . . . . .	26
Disk requirements for the active log data sets . . . . .	27
Disk requirements for the bootstrap data sets . . . . .	29
Disk requirements for the work file database . . . . .	29
Disk requirements for the default database . . . . .	32
Disk requirements for temporary table spaces . . . . .	32
Disk requirements for the dump data set size . . . . .	34
Disk requirements for the system databases . . . . .	35
Disk requirements for the archive log data sets . . . . .	35
Using the installation CLIST to calculate storage . . . . .	35
Virtual storage for address spaces . . . . .	36

DB2 distributed data facility address space (DSN1DIST)	36
IRLM address space (IRLMPROC)	36
DB2 system services address space (DSN1MSTR)	37
DB2 database services address space (DSN1DBM1)	37
Allied agent address space	37
Common service area	37
DB2-Established stored procedures address space (DSN1SPAS)	38
WLM-Established stored procedures address spaces	39
Virtual storage for storage pools and working storage	39
Buffer pool size calculation	40
Sort pool size calculation	41
RID pool size calculation	42
EDM pool size calculation	43
Data set control block storage size calculation	47
Working storage calculation	48
Virtual storage below the 16MB line	48
Real storage	49
<b>Chapter 4. Loading DB2 libraries</b>	<b>51</b>
What IBM sends you	51
What you produce	52
SMP/E step 1: Copy and edit the SMP/E jobs	55
Copying the SMP/E jobs	55
Editing the SMP/E jobs	56
SMP/E step 2: Allocate the SMP/E CSI file and SMP/E control data sets:	
DSNTIJAA (optional)	62
SMP/E step 3: Allocate distribution and target libraries: DSNALLOC	64
SMP/E step 4: Run the receive jobs: DSNRECV1, DSNRECV2, DSNRECV3, DSNRECV4	64
SMP/E step 5: Cleanup job for migration: DSNTIJUD (optional)	65
SMP/E step 6: Run the apply jobs: DSNAPPL1, DSNAPPL2	65
SMP/E step 7: Run the accept jobs: DSNACEP1, DSNACEP2	66
SMP/E step 8: Unload the jobs for the additional FMIDs (optional)	67
SMP/E step 9: Run the receive jobs for the additional FMIDs (optional)	68
SMP/E step 10: Run the apply job for the additional FMIDs (optional)	68
SMP/E step 11: Run the accept job for the additional FMIDs (optional)	68
SMP/E step 12: Copy and edit the SMP/E jobs for DB2 REXX Language Support (optional)	68
SMP/E step 13: Run the RECEIVE job for DB2 REXX Language Support (optional)	69
SMP/E step 14: Run the APPLY job for DB2 REXX Language Support (optional)	69
SMP/E step 15: Run the ACCEPT job for DB2 REXX Language Support (optional)	69
SMP/E step 16: Ensure installation of proper maintenance	69
Finishing SMP/E processing	69
<b>Chapter 5. Setting up and using DB2 Online Help</b>	<b>71</b>
Copying the bookshelf list, index, and books	71
Changing online help book data set names (optional)	71
Customizing BookManager READ for online help (optional)	72
Verifying online help and setting exit options	73
Making online help accessible from installation and DB2I panels	74
Using online help	74
Accessing help with DB2 online help	75
Moving around in the DB2 online help	75

<b>Chapter 6. Installing, migrating, and updating system parameters</b>	77
Running the installation CLIST	77
Making the DB2 ISPF libraries available to TSO.	77
Invoking the CLIST	78
General instructions	78
Directory of panels	81
Directory of panel field names	82
Online book data set names panel: DSNTIPA0	89
Main panel: DSNTIPA1	90
Recommended approach for a new installer	95
Data parameters panel: DSNTIPA2	96
Define group or member panel: DSNTIPK	101
System resource data set names: DSNTIPH	103
Data set names panel 1: DSNTIPT	107
Data set names panel 2: DSNTIPU	112
Data set names panel 3: DSNTIPQ	120
Data set names panel 4: DSNTIPG	123
Data set names panel 5: DSNTIPW	127
Sizes panel 1: DSNTIPD	130
Sizes panel 2: DSNTIP7	136
Thread management panel: DSNTIPE	138
Buffer pool sizes panel 1: DSNTIP1	143
Buffer pool sizes panel 2: DSNTIP2	145
Buffer pool sizes panel 3: DSNTIP6	147
Tracing panel: DSNTIPN	149
Operator functions panel: DSNTIPO	153
Application programming defaults panel 1: DSNTIPF	160
Application programming defaults panel 2: DSNTIP4	168
IRLM panel 1: DSNTIPI	174
IRLM panel 2: DSNTIPJ	179
Protection panel: DSNTIPP	184
MVS parmlib updates panel: DSNTIPM	188
Active log data set parameters: DSNTIPL	192
Archive log data set parameters: DSNTIPA	197
Databases and spaces to start automatically panel: DSNTIPS	203
Distributed data facility: DSNTIPR	205
Distributed data facility panel: DSNTIP5	210
Routine parameters panel: DSNTIPX	214
Data definition control support panel: DSNTIPZ	216
Job editing panel: DSNTIPY	219
Install DB2 - CLIST calculations panel 1: DSNTIPC	222
Install DB2 - CLIST calculations panel 2: DSNTIPC1	227
Completing the CLIST processing	227
Responding to messages	227
Tailoring the installation jobs	228
Editing the subsystem parameters and DSNHDECP values	231
The update process	231
Updating parameters through the update selection menu panel: DSNTIPB	231
Updating other parameters	233
<b>Chapter 7. Installing the DB2 subsystem</b>	235
Installation step 1: Define DB2 to MVS: DSNTIJMV	235
DSNTIJMV updates to SYS1.PARMLIB	236
DSNTIJMV updates to SYS1.PROCLIB	239
Installation step 2: Define the integrated catalog facility catalog and Alias:	
DSNTIJCA	240

	Installation step 3: Define system data sets: DSNTIJIN. . . . .	240
	Installation step 4: Initialize system data sets: DSNTIJID . . . . .	241
	Installation step 5: Define DB2 initialization parameters: DSNTIJUZ . . . . .	242
	Installation step 6: Define user exit routines: DSNTIJEX (optional) . . . . .	244
	Installation step 7: Record DB2 data to SMF (optional). . . . .	244
	Installation step 8: Establish subsystem security (optional) . . . . .	246
	Installation Step 9: Connect DB2 to TSO . . . . .	246
	Making DB2 load modules available to TSO and batch users . . . . .	246
	Making DB2 CLISTs available to TSO and batch users: DSNTIJVC . . . . .	246
	Ensuring that PL/I options are available . . . . .	247
	Making panels, messages, and load modules available to ISPF and TSO . . . . .	247
	Connecting DB2I panels to the ISPF main panel . . . . .	249
	Establishing DB2 authorization IDs in TSO and RACF . . . . .	253
	Installation step 10: Connect IMS to DB2 (optional) . . . . .	253
	Installation step 11: Connect CICS to DB2 (optional) . . . . .	253
	Installation step 12: IPL MVS . . . . .	253
	Installation step 13: Start the DB2 subsystem . . . . .	254
	Installation step 14: Define temporary work files: DSNTIJTM. . . . .	255
	Installation Step 15: Define and bind DB2 objects and user-maintained databases: DSNTIJSG. . . . .	257
	Installation step 16: Populate the user-maintained databases (optional). . . . .	259
	Installation step 17: Bind the packages for DB2 REXX Language Support: DSNTIJRX (optional) . . . . .	259
	Installation Step 18: Back up the DB2 directory and catalog: DSNTIJIC . . . . .	259
	Authorizing DB2 users. . . . .	260
	Installation step 19: Verify the installation. . . . .	260
	Installing support for a communications network . . . . .	260
	Installing a second DB2 on the same operating system . . . . .	261
	Planning considerations . . . . .	261
	Installation procedure . . . . .	262
	Considerations for using shared DASD . . . . .	266
#	Loading data with an SQL cursor. . . . .	266
	Enabling stored procedures after installation . . . . .	266
	Enabling DB2-supplied stored procedures . . . . .	268
	Enabling Control Center procedures . . . . .	268
	Enabling SQL procedure support. . . . .	269
#	Enabling CICS transaction invocation procedure . . . . .	269
	<b>Chapter 8. Migrating from Version 5 to Version 7. . . . .</b>	<b>271</b>
	Migration Considerations. . . . .	271
	Scrollable cursors are supported . . . . .	271
	Unicode support . . . . .	271
	Encoding schemes of string parameters for routines. . . . .	271
	Revoking SYSADM authorization. . . . .	272
	Changed messages from the LOAD utility . . . . .	272
	Change in recording soft errors in SYS1.LOGREC data set . . . . .	272
	More than 32 K databases are supported. . . . .	272
	Support for up to 150 000 connections. . . . .	272
	Log buffer size increased . . . . .	272
	Increased maximum number of data sets open . . . . .	272
	Changes to the RLST . . . . .	273
	Creating tables with DBCS and mixed columns . . . . .	273
	Increased size for generated code . . . . .	273
	Consider increasing IDBACK and CTHREAD . . . . .	273
	Customized DB2I defaults can be migrated . . . . .	273
	Migrating a data sharing group . . . . .	273



	Consider enlarging BSDS . . . . .	274
	Stored procedures . . . . .	274
	Utility enhancements . . . . .	275
	Work file database size calculations. . . . .	275
	Data set password protection is removed. . . . .	275
	SYSIBM.SYSPROCEDURES no longer used . . . . .	276
	Private protocol function not enhanced. . . . .	276
	Type 2 indexes are required . . . . .	276
	Shared read-only data is removed . . . . .	276
I	DCE security authentication no longer supported . . . . .	276
	Release coexistence . . . . .	276
	Migration step 1: Premigration activities . . . . .	278
I	Remove incomplete table definitions . . . . .	278
	Identify unsupported objects (DSNTIJPM) . . . . .	279
	Save critical access paths (optional) . . . . .	280
	DRDA support for three-part names (optional) . . . . .	281
	Examine all new and changed values for DB2I panels . . . . .	281
	Make adjustments for release incompatibilities . . . . .	281
	Migration Step 2: Run link checker on DB2 for OS/390 Version 5 table spaces (optional). . . . .	286
	Migration step 3: Determine which plans and packages are invalid after migration (optional) . . . . .	287
	Migration step 4: Check for consistency between catalog tables (optional) . . . . .	288
	Migration step 5: Image copy directory and catalog in case of fallback: DSNTIJIC . . . . .	288
	Migration step 6: Connect DB2 to TSO . . . . .	289
	Making DB2 load modules available to TSO and batch users . . . . .	290
	Making DB2 CLISTs available to TSO and batch users: DSNTIJVC . . . . .	290
	Making panels, messages, and load modules available to ISPF and TSO . . . . .	291
	Migration step 7: Connect IMS to DB2 (optional) . . . . .	292
	Migration step 8: Connect CICS to DB2 (optional) . . . . .	292
	Migration step 9: Stop DB2 Version 5 activity . . . . .	292
	Migration step 10: Back up your DB2 for OS/390 Version 5 volumes (optional) . . . . .	293
	Migration step 11: Define DB2 initialization parameters: DSNTIJUZ . . . . .	293
	DSNTIJUZ actions . . . . .	294
	Additional steps . . . . .	294
	Considerations for the BSDS . . . . .	294
	Migration step 12: Establish subsystem security (optional) . . . . .	295
	Migration step 13: Define DB2 Version 7 to MVS: DSNTIJMV . . . . .	295
	DSNTIJMV actions . . . . .	295
	Completing the step . . . . .	297
	Migration step 14: Define system data sets: DSNTIJIN. . . . .	297
	Migration step 15: Define user authorization exit routines: DSNTIJEX . . . . .	297
	Migration step 16: IPL MVS. . . . .	298
	Migration step 17: Start DB2 for OS/390 and z/OS Version 7 . . . . .	299
	Migration step 18: Tailor DB2 for OS/390 and z/OS Version 7 catalog: DSNTIJTC . . . . .	300
	Migration step 19: Ensure there are no problems with the catalog (optional) . . . . .	301
	Migration step 20: Prepare dynamic SQL program: DSNTIJTM. . . . .	301
	Migration step 21: Bind SPUFI and DCLGEN and user maintained database activity: DSNTIJSG . . . . .	301
I	Migration step 22: Migrate SQL procedures tables: DSNTIJMP (optional) . . . . .	302
	Migration step 23: Bind the packages for DB2 REXX Language Support: DSNTIJRX (optional) . . . . .	302
	Migration step 24: Image copy DB2 for OS/390 and z/OS Version 7 catalog: DSNTIJIC . . . . .	302

Migration step 25: Verify your DB2 for OS/390 and z/OS Version 7 system	303
Running the DB2 Version 5 sample jobs	303
Running the Version 7 sample jobs	303
Testing Version 7 with your application test cases	304
Falling back	304
Fallback considerations	304
Fallback procedure	308
Fallback Step 1: Run phase 0 of the DB2 for OS/390 and z/OS Version 7 installation verification procedure	309
Fallback Step 2: Stop DB2 for OS/390 and z/OS Version 7 activity	309
Fallback Step 3: Reactivate DB2 Version 5 code: DSNTIJFV	310
Fallback Step 4: Reconnect TSO, IMS, and CICS to DB2 Version 5	310
Fallback Step 5: Start DB2 Version 5	310
Fallback Step 6: Verify fallback	312
Remigrating	312
<b>Chapter 9. Migrating from Version 6 to Version 7.</b>	315
Migration Considerations	315
Scrollable cursors are supported	315
Unicode support	315
New default encoding scheme	316
Revoking SYSADM authorization	316
Encoding schemes of string parameters for routines	316
Changed messages from the LOAD utility	316
Creating tables with DBCS and mixed columns	316
Increased size for generated code	316
Consider increasing IDBACK and CTHREAD	316
Change in recording soft errors in SYS1.LOGREC data set	316
Customized DB2I defaults can be migrated	317
Migrating a data sharing group	317
Work file database size calculations	317
SYSIBM.SYSPSM and SYSIBM.SYSPSMOPTS no longer used	318
Release coexistence	318
Migration step 1: Premigration activities	319
Remove incomplete table definitions	319
Save critical access paths (optional)	320
Examine all new and changed values for DB2I panels	321
DRDA support for three-part names (optional)	321
Make adjustments for release incompatibilities	321
Migration Step 2: Run link checker on DB2 for OS/390 Version 6 table spaces (optional)	324
Migration step 3: Determine which plans and packages are invalid after migration (optional)	325
Migration step 4: Check for consistency between catalog tables (optional)	325
Migration step 5: Image copy directory and catalog in case of fallback: DSNTIJIC	326
Migration step 6: Connect DB2 to TSO	327
Making DB2 load modules available to TSO and batch users	327
Making DB2 CLISTs available to TSO and batch users: DSNTIJVC	327
Making panels, messages, and load modules available to ISPF and TSO	328
Migration step 7: Connect IMS to DB2 (optional)	329
Migration step 8: Connect CICS to DB2 (optional)	329
Migration step 9: Stop DB2 Version 6 activity	330
Migration step 10: Back up your DB2 for OS/390 Version 6 volumes (optional)	331
Migration step 11: Define DB2 initialization parameters: DSNTIJUZ DSNTIJUZ actions	331

Additional steps . . . . .	331
Considerations for the BSDS . . . . .	332
Migration step 12: Establish subsystem security (optional) . . . . .	332
Migration step 13: Define DB2 Version 7 to MVS: DSNTIJMV . . . . .	332
DSNTIJMV actions . . . . .	333
Completing the step . . . . .	334
Migration step 14: Define system data sets: DSNTIJIN . . . . .	335
Migration step 15: Define user authorization exit routines: DSNTIJEX . . . . .	335
Migration step 16: IPL MVS . . . . .	336
Migration step 17: Start DB2 for OS/390 and z/OS Version 7 . . . . .	336
Migration step 18: Tailor DB2 for OS/390 and z/OS Version 7 catalog:	
DSNTIJTC . . . . .	337
Migration step 19: Ensure there are no problems with the catalog (optional) . . . . .	338
Migration step 20: Prepare dynamic SQL program: DSNTIJTM . . . . .	338
Migration step 21: Bind SPUFI and DCLGEN and user maintained database	
activity: DSNTIJSG . . . . .	339
Migration step 22: Migrate SQL procedures tables: DSNTIJMP (optional) . . . . .	340
Migration step 23: Bind the packages for DB2 REXX Language Support:	
DSNTIJRX (optional) . . . . .	340
Migration step 24: Image copy DB2 for OS/390 and z/OS Version 7 catalog:	
DSNTIJIC . . . . .	340
Migration step 25: Verify your DB2 for OS/390 and z/OS Version 7 system . . . . .	340
Running the DB2 Version 6 sample jobs . . . . .	340
Running the Version 7 sample jobs . . . . .	341
Testing Version 7 with your application test cases . . . . .	341
Falling back . . . . .	341
Fallback considerations . . . . .	342
Fallback procedure . . . . .	344
Fallback Step 1: Run phase 0 of the DB2 for OS/390 and z/OS Version 7	
installation verification procedure . . . . .	344
Fallback Step 2: Stop DB2 for OS/390 and z/OS Version 7 activity . . . . .	344
Fallback Step 3: Reactivate DB2 Version 6 code: DSNTIJFV . . . . .	345
Fallback Step 4: Reconnect TSO, IMS, and CICS to DB2 Version 6 . . . . .	345
Fallback Step 5: Start DB2 Version 6 . . . . .	346
Fallback Step 6: Verify fallback . . . . .	347
Remigrating . . . . .	347
<b>Chapter 10. Verifying with the sample applications . . . . .</b>	<b>349</b>
Installation verification phases and programs . . . . .	350
Planning for verification . . . . .	354
Special considerations for COBOL programs . . . . .	355
Special considerations for C and C++ programs . . . . .	357
Special considerations for PL/I programs . . . . .	358
Phase 0: Deleting the sample objects (DSNTEJ0) . . . . .	359
Phase 1: Creating and loading sample tables . . . . .	360
Job DSNTEJ1 . . . . .	360
Job DSNTEJ1L . . . . .	362
Job DSNTEJ1P . . . . .	363
Job DSNTEJ1U . . . . .	364
Phase 2: Testing the batch environment . . . . .	364
Job DSNTEJ2A . . . . .	364
Job DSNTEJ2C . . . . .	365
Job DSNTEJ2D . . . . .	365
Job DSNTEJ2E . . . . .	366
Job DSNTEJ2F . . . . .	366
Job DSNTEJ2P . . . . .	367

Job DSNTEJ2U . . . . .	367
Phase 3: Testing SPUFI, DRDA Access, dynamic SQL, and TSO . . . . .	369
Testing SPUFI. . . . .	370
Running SPUFI at remote non-DB2 systems . . . . .	370
Running dynamic SQL and the ISPF/CAF application . . . . .	371
Jobs DSNTEJ3C and DSNTEJ3P: . . . . .	371
Starting an application in an ISPF/TSO environment. . . . .	373
Phase 4: Testing the IMS environment. . . . .	373
Jobs DSNTEJ4C and DSNTEJ4P . . . . .	374
Starting an application in an IMS environment . . . . .	376
Using the phone application in IMS . . . . .	377
Phase 5: Testing the CICS environment . . . . .	377
Job DSNTEJ5A . . . . .	377
Jobs DSNTEJ5C and DSNTEJ5P . . . . .	377
Starting an application in a CICS environment . . . . .	380
Using the phone application in CICS . . . . .	380
Using CICS storage-handling facilities . . . . .	381
Phase 6: Accessing data at a remote site. . . . .	381
DRDA Access sample . . . . .	382
Job DSNTEJ6. . . . .	382
DB2 Private Protocol Access sample . . . . .	383
Stored procedure samples . . . . .	384
Stored procedure sample without result set . . . . .	384
Job DSNTEJ6S . . . . .	385
Job DSNTEJ6P . . . . .	385
Stored procedure sample with result set . . . . .	386
Job DSNTEJ6T . . . . .	386
Job DSNTEJ6D . . . . .	386
Sample utilities stored procedure . . . . .	387
Job DSNTEJ6U . . . . .	387
Job DSNTEJ6V . . . . .	388
Job DSNTEJ6W . . . . .	388
Job DSNTEJ6Z . . . . .	389
Sample ODBA stored procedure . . . . .	389
Job DSNTEJ61 . . . . .	390
Job DSNTEJ62 . . . . .	390
Sample SQL procedures . . . . .	390
Job DSNTEJ63 . . . . .	391
Job DSNTEJ64 . . . . .	391
Job DSNTEJ65 . . . . .	392
Starting an application in an ISPF/TSO environment in phase 6 . . . . .	392
Phase 7: Accessing LOB Data. . . . .	393
Job DSNTEJ7. . . . .	393
Job DSNTEJ71 . . . . .	394
Job DSNTEJ73 . . . . .	394
Job DSNTEJ75 . . . . .	395
Starting an application in an ISPF/TSO environment in phase 7 . . . . .	395
The sample applications . . . . .	395
Printing the sample application listings. . . . .	396
Specifying values in the sample application panels . . . . .	396
Scenarios . . . . .	399
The project application scenario . . . . .	399
The organization application scenario . . . . .	401
The phone application scenario . . . . .	405
The distributed organization application scenario . . . . .	409
Employee resume and photo scenarios . . . . .	417

Edit exit routine . . . . .	420
Huffman compression exit routine . . . . .	421
Sample field procedure . . . . .	421
Dynamic SQL statements (DSNTESA, DSNTESQ) . . . . .	421
DSNTESA . . . . .	421
DSNTESQ . . . . .	423
Dynamic SQL programs (DSNTIAD, DSNTPE2, DSNTIAUL) . . . . .	423
<b>Chapter 11. Connecting the IMS attachment facility . . . . .</b>	<b>425</b>
Making DB2 load modules available to IMS . . . . .	425
Defining DB2 to IMS . . . . .	425
Defining new programs and transactions to IMS . . . . .	429
Defining DB2 plans for IMS applications (optional) . . . . .	429
Generating a user language interface (optional) . . . . .	430
IMS attachment facility macro (DSNMAPN) . . . . .	431
Description . . . . .	431
<b>Chapter 12. Connecting the CICS attachment facility . . . . .</b>	<b>433</b>
Using the attachment facility for CICS Version 4 and CICS TS Version 1.1 . . . . .	433
Calculating space requirements for the CICS attachment facility . . . . .	433
CICS Version 4 . . . . .	434
Estimating storage needed for CICS attachment threads . . . . .	435
Defining the DB2 to CICS connection using the RCT . . . . .	436
Updating the CICS system tables . . . . .	436
System initialization table entries . . . . .	436
Transaction entries . . . . .	437
Program entries . . . . .	439
PLT processing . . . . .	440
Providing CICS support for DB2 commands . . . . .	441
Updating CICS initialization JCL . . . . .	442
Coordinating DB2 and CICS security . . . . .	442
Preparing CICS applications for DB2 . . . . .	443
CICS attachment facility macro (DSNCRCT) . . . . .	444
DSNCRCT TYPE=INIT . . . . .	444
DSNCRCT TYPE=COMD . . . . .	451
DSNCRCT TYPE=POOL . . . . .	452
DSNCRCT TYPE=ENTRY . . . . .	453
DSNCRCT TYPE=FINAL . . . . .	461
DSNCRCT TYPE=DSECT . . . . .	461

---

## **Part 3. Communicating with other systems. . . . . 463**

<b>Chapter 13. Connecting distributed database systems . . . . .</b>	<b>465</b>
The database protocols (DRDA vs private) . . . . .	465
The communications protocols . . . . .	465
The role of the communications database (CDB) . . . . .	466
Enhancements since Version 5 . . . . .	467
DB2 installation considerations . . . . .	467
<b>Chapter 14. Connecting systems with VTAM . . . . .</b>	<b>469</b>
Step 1: Customize VTAM for DB2 . . . . .	471
Step 2: Choose names and a password . . . . .	471
Names for the local subsystem . . . . .	471
A password for the local subsystem . . . . .	472
Names you need from the remote systems . . . . .	472
Names chosen by Spiffy Computer Company . . . . .	473

Step 3: Define the DB2 subsystem to VTAM . . . . .	473
The APPL statement . . . . .	473
The modeent macro . . . . .	477
Step 4: Populate the communications database . . . . .	480
SYSIBM.LOCATIONS table . . . . .	480
SYSIBM.LUNAMES table . . . . .	481
SYSIBM.USERNAMES table . . . . .	483
Step 5: Start VTAM to use DB2 . . . . .	483
Step 6: Tune the system . . . . .	484
Controlling buffer storage . . . . .	485
Controlling pacing . . . . .	486
Modifying default session limits . . . . .	487
Modifying class of service . . . . .	488
Associating applications with modes . . . . .	488
Updating CDB values . . . . .	491
Calculating session limits. . . . .	492
Calculating VTAM I/O buffer pool (IOBUF) storage . . . . .	495
Interpreting CNOS messages . . . . .	496
Sample VTAM definitions to connect two DB2s . . . . .	499
Basic definitions . . . . .	499
Channel-connected DB2s . . . . .	502
NCP-connected DB2s . . . . .	503
Using the change log inventory utility to update the BSDS . . . . .	507
 <b>Chapter 15. Connecting systems with TCP/IP . . . . .</b>	 509
Enabling TCP/IP communication . . . . .	510
TCP/IP limitations . . . . .	510
Initializing a TCP stack for use with a VIPA . . . . .	511
Using two-phase commit . . . . .	511
Multiple DB2 subsystems on a single OS/390 image . . . . .	511
Step 1: Prepare the Language Environment runtime library . . . . .	512
Step 2: Enable DDF for UNIX System Services . . . . .	513
Step 3: Define the DB2 subsystem to TCP/IP . . . . .	513
Customize the TCP/IP data sets or files . . . . .	514
Modify the change log inventory job. . . . .	515
Step 4: Populate the communications database . . . . .	516
SYSIBM.LOCATIONS table . . . . .	516
SYSIBM.IPNAMES table . . . . .	517
SYSIBM.USERNAMES table . . . . .	518
Step 5: Start TCP/IP support . . . . .	518
Step 6: Tuning TCP/IP. . . . .	519

---

## Part 4. Appendixes . . . . . 521

<b>Appendix A. Character conversion . . . . .</b>	<b>523</b>
Understanding character conversion . . . . .	523
Specifying a system coded character set identifier . . . . .	524
Unicode support . . . . .	527
Special considerations. . . . .	527
Converting to the euro symbol. . . . .	528
How an entry in SYSIBM.SYSSTRINGS works. . . . .	529
Rules for SYSSTRINGS entries . . . . .	531
When remote packages should be rebound . . . . .	531
Specifying locales . . . . .	531
 <b>Appendix B. Directory of subsystem parameters . . . . .</b>	 533

	Editing the subsystem parameters and DSNHDECP values . . . . .	533
	Directory of subsystem parameters and DSNHDECP values. . . . .	533
	<b>Appendix C. DB2 utilities packaging.</b> . . . .	537
	SMP/E jobs for DB2 utility products . . . . .	538
	Installing DB2 Operational Utilities . . . . .	538
	Installing DB2 Diagnostic & Recovery Utilities . . . . .	538
	Installing DB2 Utilities Suite. . . . .	539
	How to operate in a mixed-release data sharing environment . . . . .	539
	<b>Notices</b> . . . . .	541
	Programming interface information . . . . .	542
	Trademarks. . . . .	543
	<b>Glossary</b> . . . . .	545
	<b>Bibliography</b> . . . . .	565
	<b>Index</b> . . . . .	571





---

## About this book

This chapter contains specific information about this book and a general overview of the DB2 Universal Database™ for OS/390® and z/OS library.

### Important

In this version of DB2® for OS/390 and z/OS, some utility functions are available as optional products. You must separately order and purchase a license to such utilities, and discussion of those utility functions in this publication is not intended to otherwise imply that you have a license to them.

---

## 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.

DB2 Installer is feature of DB2 that provides a graphical user interface for customizing DB2 for OS/390 and z/OS jobs on the workstation. DB2 Installer provides an alternative to the existing ISPF installation panels and CLISTs. DB2 Installer lets you install, migrate, and update your DB2 subsystem. If you plan to install DB2 from your workstation using DB2 Installer, refer to the instructions that come with DB2 Installer. You can download DB2 Installer from the DB2 for OS/390 and z/OS web site.

This book assumes that you are 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 to use the DB2 library

Titles of books in the library begin with DB2 Universal Database for OS/390 and z/OS Version 7. 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.

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 that are associated with DB2 are grouped into the following major categories (but supplemental information relating to all of the following tasks for new releases of DB2 can be found in *DB2 Release Planning 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 capabilities you also need *DB2 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 *An Introduction to DB2 for OS/390*, *DB2 Application Programming and SQL Guide*, and *DB2 SQL Reference*.

End users can also issue SQL statements through the Query Management Facility (QMF™) or some other program, and the library for that licensed program might provide all the instruction or reference material they need. For a list 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 *DB2 Application Programming and SQL Guide*, *DB2 SQL Reference*, and *DB2 ODBC 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 operating systemss
- How to write applications that use DB2 ODBC 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 *DB2 Application Programming and SQL Guide* and in *DB2 Application Programming Guide and Reference for Java*. The material needed for writing applications that use DB2 ODBC or ODBC to access DB2 servers is in *DB2 ODBC Guide and Reference*. For handling errors, see *DB2 Messages and Codes*.

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

**System and Database Administration:** *Administration* covers almost everything else. *DB2 Administration Guide* divides those tasks among the following sections:

- Part 2 (Volume 1) of *DB2 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.
- Part 3 (Volume 1) of *DB2 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.
- Part 4 (Volume 1) of *DB2 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.
- Part 5 (Volume 2) of *DB2 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 *DB2 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 *An Introduction to DB2 for OS/390* and *DB2 Administration Guide*. If you also want to carry out your own plans by creating DB2 objects, granting privileges, running utility jobs, and so on, you also need:

- *DB2 SQL Reference*, which describes the SQL statements you use to create, alter, and drop objects and grant and revoke privileges
- *DB2 Utility Guide and Reference*, which explains how to run utilities
- *DB2 Command Reference*, which explains how to run commands

If you will be using data sharing, you need *DB2 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 *DB2 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 *DB2 Diagnosis Guide and Reference* and *DB2 Messages and Codes*.

---

## How to obtain DB2 information

This section provides information that you can use to find valuable information about the DB2 product.

### DB2 on the Web

Stay current with the latest information about DB2. View the DB2 home page on the 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.

You can view and search DB2 publications on the Web, or you can download and print many of the most current DB2 books. Follow links to other Web sites with more information about DB2 family and OS/390 solutions. Access DB2 on the Web at the following address:

[www.ibm.com/software/db2os390](http://www.ibm.com/software/db2os390)

## DB2 publications

The publications for DB2 for OS/390 and z/OS are available in softcopy format and hardcopy. IBM provides mid-version updates in softcopy on the Web and on CD-ROM.

### BookManager format

You can use online books on CD-ROM to read, search across books, print portions of the text, and make notes in these BookManager books. Using the IBM Softcopy Reader, appropriate IBM Library Readers, or the BookManager Read product, you can view these books in the OS/390, VM, and Windows environments. You can also view and search many of the DB2 BookManager books on the Web.

### PDF format

Many of the DB2 books are available in Portable Document Format (PDF) for viewing or printing from CD-ROM or the Web. Download the PDF books to your intranet for distribution throughout your enterprise.

### CD-ROMs and DVD

Books for Version 7 of DB2 Universal Database Server for OS/390 and z/OS are available on a CD-ROM that is included with your shipment of DB2 for OS/390:

- *DB2 UDB Server for OS/390 Version 7 Library Collection*, LK3T-6999, in English

The CD-ROM contains the collection of books for the DB2 server in BookManager and PDF formats. Periodically, IBM refreshes the books on subsequent editions of this CD-ROM.

The books for Version 7 of DB2 UDB Server for OS/390 and z/OS are also available on the following CD-ROM and DVD collection kits, which contain online books for many IBM products:

- *z/OS Software Collection*, SK3T-4270, in English
- *z/OS and Software Products DVD Collection*, SK3T-4271-00, in English
- *Online Library Omnibus Edition OS/390 Collection*, SK2T-6700, in English

## DB2 education

IBM Education and Training offers a wide variety of classroom courses to help you quickly and efficiently gain DB2 expertise. IBM schedules classes in cities all over the world. You can find class information, by country, at the IBM Learning Services Web site:

[www.ibm.com/services/learning/](http://www.ibm.com/services/learning/)

For more information, including the current local schedule, please contact your IBM representative.

IBM also offers classes at your location, at a time that suits your needs. IBM can customize courses to meet your exact requirements. The *All-in-One Education and Training Catalog* describes the DB2 curriculum in the United States. You can inquire about or enroll in these courses by calling 1-800-IBM-TEACH (1-800-426-8322).

## How to order the DB2 library

You can order DB2 publications and CD-ROMs through your IBM representative or the IBM branch office that serves your locality. If your location is 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 option. Be prepared to give your customer number, the product number, and either the feature codes or order numbers that you want.

You can also order books from the IBM Publication Center on the Web:  
[www.elink.ibm.link.ibm.com](http://www.elink.ibm.link.ibm.com)

## Subscription through the Publication Notification System (PNS)

IBM has replaced the System Library Subscription Service (SLSS) with an up-to-date notification application, the Publication Notification System (PNS). IBM migrated all active SLSS subscriptions to the new PNS application, which you can access from the Web.

PNS users receive electronic notifications of updated publications in their profiles. You have the option of ordering the updates by using the publications direct ordering application or any other IBM publication ordering channel. Unlike SLSS, the PNS application does not send automatic shipments of publications. You will receive updated publications and a bill for them if you respond to the electronic notification. To access the PNS application on the World Wide Web, enter the following address on your Web browser command line:

[www.ibm.com/shop/publications/pns/elink.ibm.link.ibm.com](http://www.ibm.com/shop/publications/pns/elink.ibm.link.ibm.com)

---

## Product terminology and citations

In this book, DB2 Universal Database Server for OS/390 and z/OS is referred to as "DB2 for OS/390 and z/OS." In cases where the context makes the meaning clear, DB2 for OS/390 and z/OS is referred to as "DB2." When this book refers to other books in this library, a short title is used. (For example, "See *DB2 SQL Reference*" is a citation to *IBM DATABASE 2™ Universal Database Server for OS/390 and z/OS SQL Reference*.)

When referring to a DB2 product other than DB2 for OS/390 and z/OS, this book uses the product's full name to avoid ambiguity.

The following terms are used as indicated:

**DB2** Represents either the DB2 licensed program or a particular DB2 subsystem.

**C and C language**

Represent the C programming language.

**CICS** Represents CICS/ESA® and CICS Transaction Server for OS/390.

**IMS** Represents IMS or IMS/ESA®.

**MVS** Represents the MVS element of OS/390.

**OS/390**

Represents the OS/390 or z/OS operating system.

**RACF®**

Represents the functions that are provided by the RACF component of the SecureWay® Security Server for OS/390 or by the RACF component of the OS/390 Security Server.

---

## How to read the syntax diagrams

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

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

The ►— symbol indicates the beginning of a statement.

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

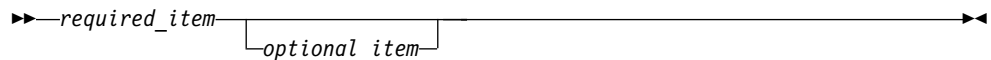
The ►— symbol indicates that a statement is continued from the previous line.

The —► symbol indicates the end of a statement.

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



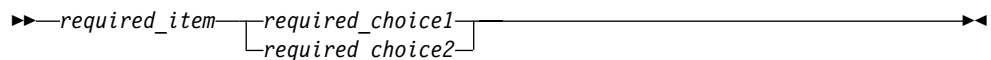
- Optional items appear below the main path.



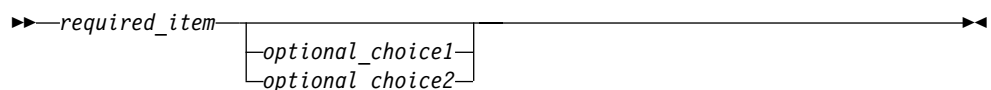
If an optional item appears above the main path, that item has no effect on the execution of the statement and is used only for readability.



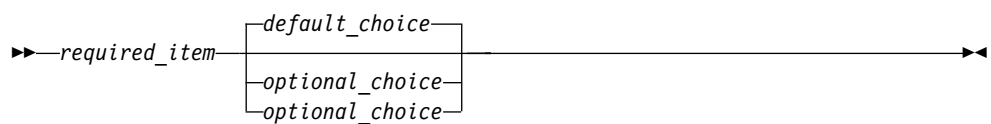
- If you can choose from two or more items, they appear vertically, in a stack. If you *must* choose one of the items, one item of the stack appears on the main path.



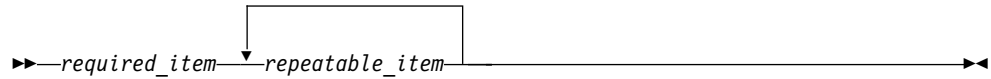
If choosing one of the items is optional, the entire stack appears below the main path.



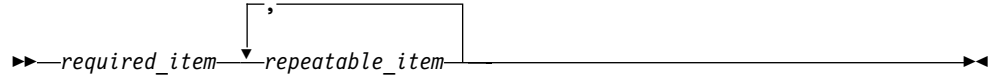
If one of the items is the default, it appears above the main path and the remaining choices are shown below.



- 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 send your comments

Your feedback helps IBM to provide quality information. Please send any comments that you have about this book or other DB2 for OS/390 and z/OS documentation. You can use any of the following methods to provide comments:

- Send your comments by e-mail to [db2pubs@vnet.ibm.com](mailto:db2pubs@vnet.ibm.com) and include the name of the product, the version number of the product, and the number of the book. If you are commenting on specific text, please list the location of the text (for example, a chapter and section title, page number, or a help topic title).
- Send your comments from the Web. Visit the Web site at:

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

The Web site has a feedback page that you can use to send comments.

- Complete the readers' comment form at the back of the book and return it by mail, by fax (800-426-7773 for the United States and Canada), or by giving it to an IBM representative.





---

## Summary of changes to this book

This section summarizes the changes that have been made to this book.

### **Section 2. Planning and installing DB2 has the following changes:**

- Chapter 2. Introduction to installation and migration contains updated summaries of installation and migration steps.
- Chapter 3. Estimating DB2 storage needs provides calculations to help you estimate storage for the DB2 subsystem.
- Chapter 6. Installing, migrating, and updating system parameters describes new panel fields to support thread management, application programming defaults, Unicode, and other enhancements.
- Chapter 9. Migrating from Version 6 to Version 7 includes a new, optional, migration step.
- Chapter 8. Migrating from Version 5 to Version 7 and Chapter 9. Migrating from Version 6 to Version 7 describe migration considerations for migrating to Version 7 and the migration process.
- Chapter 8. Migrating from Version 5 to Version 7 and Chapter 9. Migrating from Version 6 to Version 7 now contain information about falling back.
- Chapter 10. Verifying with the sample applications includes information on four new sample jobs.

### **Section 3. Communicating with other systems has the following changes:**

- Chapter 13. Connecting distributed database systems has minor changes for Version 7 enhancements.
- Chapter 14. Connecting systems with VTAM has minor changes for LU names.
- Chapter 15. Connecting systems with TCP/IP has minor changes for LU names and general changes to TCP/IP.

### **The appendixes have the following changes:**

- Appendix A. Character conversion includes information about using Unicode.
- Appendix B. Directory of subsystem parameters has been moved from Chapter 6. Installing, migrating, and updating system parameters. Notices have been moved to Appendix C.
- Appendix C. DB2 utilities packaging is a new appendix describing installation for the DB2 Utilities Suite.



---

## Part 1. Introduction

<b>Chapter 1. Summary of changes to DB2 for OS/390 and z/OS Version 7.</b>	3
Enhancements for managing data	3
Enhancements for reliability, scalability, and availability.	3
Easier development and integration of e-business applications.	4
Improved connectivity	5
Features of DB2 for OS/390 and z/OS.	6
Migration considerations	6
 <b>Chapter 2. Introduction to installation and migration</b>	 7
Installing other features of the DB2 UDB Server for OS/390 and z/OS	7
Installation features.	7
Installation and migration steps overview.	8
Planning for migration incompatibilities	9
SMP/E steps summary	9
Installation steps summary	10
Migration steps summary	10
Fallback steps summary	12
Remigration steps summary	12



---

## Chapter 1. Summary of changes to DB2 for OS/390 and z/OS Version 7

DB2 for OS/390 and z/OS Version 7 delivers an enhanced relational database server solution for OS/390. This release focuses on greater ease and flexibility in managing your data, better reliability, scalability, and availability, and better integration with the DB2 family.

In Version 7, some utility functions are available as optional products; you must separately order and purchase a license to such utilities. Discussion of utility functions in this publication is not intended to otherwise imply that you have a license to them. See *DB2 Utility Guide and Reference* for more information about utilities products.

---

### Enhancements for managing data

Version 7 delivers the following enhancements for managing data:

- DB2 now collects a comprehensive statistics history that:
  - Lets you track changes to the physical design of DB2 objects
  - Lets DB2 predict future space requirements for table spaces and indexes more accurately and run utilities to improve performance
- Database administrators can now manage DB2 objects more easily and no longer must maintain their utility jobs (even when new objects are added) by using enhancements that let them:
  - Dynamically create object lists from a pattern-matching expression
  - Dynamically allocate the data sets that are required to process those objects
- More flexible DBADM authority lets database administrators create views for other users.
- Enhancements to management of constraints let you specify a constraint at the time you create primary or unique keys. A new restriction on the DROP INDEX statement requires that you drop the primary key, unique key, or referential constraint before you drop the index that enforces a constraint.

---

### Enhancements for reliability, scalability, and availability

Version 7 delivers the following enhancements for the reliability, scalability, and availability of your e-business:

- The DB2 Utilities Suite provides utilities for all of your data management tasks that are associated with the DB2 catalog.
- The new UNLOAD utility lets you unload data from a table space or an image copy data set. In most cases, the UNLOAD utility is faster than the DSNTIAUL sample program, especially when you activate partition parallelism for a large partitioned table space. UNLOAD is also easier to use than REORG UNLOAD EXTERNAL.
- The new COPYTOCOPY utility lets you make additional image copies from a primary image copy and registers those copies in the DB2 catalog. COPYTOCOPY leaves the target object in read/write access mode (UTRW), which allows Structured Query Language (SQL) statements and some utilities to run concurrently with the same target objects.

- Parallel LOAD with multiple inputs lets you easily load large amounts of data into partitioned table spaces for use in data warehouse applications or business intelligence applications. Parallel LOAD with multiple inputs runs in a single step, rather than in different jobs.
- A faster online REORG is achieved through the following enhancements:
  - Online REORG no longer renames data sets, which greatly reduces the time that data is unavailable during the SWITCH phase.
  - Additional parallel processing improves the elapsed time of the BUILD2 phase of REORG SHRLEVEL(CHANGE) or SHRLEVEL(REFERENCE).
- More concurrency with online LOAD RESUME is achieved by letting you give users read and write access to the data during LOAD processing so that you can load data concurrently with user transactions.
- More efficient processing for SQL queries:
  - More transformations of subqueries into a join for some UPDATE and DELETE statements
  - Fewer sort operations for queries that have an ORDER BY clause and WHERE clauses with predicates of the form *COL=constant*
  - More parallelism for IN-list index access, which can improve performance for queries involving IN-list index access
- The ability to change system parameters without stopping DB2 supports online transaction processing and e-business without interruption.
- Improved availability of user objects that are associated with failed or canceled transactions:
  - You can cancel a thread without performing rollback processing.
  - Some restrictions imposed by the restart function have been removed.
  - A NOBACKOUT option has been added to the CANCEL THREAD command.
- Improved availability of the DB2 subsystem when a log-read failure occurs: DB2 now provides a timely warning about failed log-read requests and the ability to retry the log read so that you can take corrective action and avoid a DB2 outage.
- Improved availability in the data sharing environment:
  - Group attachment enhancements let DB2 applications generically attach to a DB2 data sharing group.
  - A new LIGHT option of the START DB2 command lets you restart a DB2 data sharing member with a minimal storage footprint, and then terminate normally after DB2 frees the retained locks that it can.
  - You can let changes in structure size persist when you rebuild or reallocate a structure.
- Additional data sharing enhancements include:
  - Notification of incomplete units of recovery
  - Use of a new OS/390 and z/OS function to improve failure recovery of group buffer pools
- An additional enhancement for e-business provides improved performance with preformatting for INSERT operations.

---

## Easier development and integration of e-business applications

Version 7 provides the following enhancements, which let you more easily develop and integrate applications that access data from various DB2 operating systems and distributed environments:

- DB2 XML Extender for OS/390 and z/OS, a new member of the DB2 Extender family, lets you store, retrieve, and search XML documents in a DB2 database.

- Improved support for UNION and UNION ALL operators in a view definition, a nested table expression, or a subquery predicate, improves DB2 family compatibility and is consistent with SQL99 standards.
- More flexibility with SQL gives you greater compatibility with DB2 on other operating systems:
  - Scrollable cursors let you move forward, backward, or randomly through a result table or a result set. You can use scrollable cursors in any DB2 applications that do not use DB2 private protocol access.
  - A search condition in the WHERE clause can include a subquery in which the base object of both the subquery and the searched UPDATE or DELETE statement are the same.
  - A new SQL clause, FETCH FIRST *n* ROWS, improves performance of applications in a distributed environment.
  - Fast implicit close in which the DB2 server, during a distributed query, automatically closes the cursor when the application attempts to fetch beyond the last row.
  - Support for options USER and USING in a new authorization clause for CONNECT statements lets you easily port applications that are developed on the workstation to DB2 for OS/390 and z/OS. These options also let applications that run under WebSphere to reuse DB2 connections for different users and to enable DB2 for OS/390 and z/OS to check passwords.
  - For positioned updates, you can specify the FOR UPDATE clause of the cursor SELECT statement without a list of columns. As a result, all updatable columns of the table or view that is identified in the first FROM clause of the fullselect are included.
  - A new option of the SELECT statement, ORDER BY *expression*, lets you specify operators as the sort key for the result table of the SELECT statement.
  - New datetime ISO functions return the day of the week with Monday as day 1 and every week with seven days.
- Enhancements to Open Database Connectivity (ODBC) provide partial ODBC 3.0 support, including many new application programming interfaces (APIs), which increase application portability and alignment with industry standards.
- Enhancements to the LOAD utility let you load the output of an SQL SELECT statement directly into a table.
- A new component called Precompiler Services lets compiler writers modify their compilers to invoke Precompiler Services and produce an *SQL statement coprocessor*. An SQL statement coprocessor performs the same functions as the DB2 precompiler, but it performs those functions at compile time. If your compiler has an SQL statement coprocessor, you can eliminate the precompile step in your batch program preparation jobs for COBOL and PL/I programs.
- Support for Unicode-encoded data lets you easily store multilingual data within the same table or on the same DB2 subsystem. The Unicode encoding scheme represents the code points of many different geographies and languages.

#  
#  
#  
#  
#  
#

---

## Improved connectivity

Version 7 offers improved connectivity:

- Support for COMMIT and ROLLBACK in stored procedures lets you commit or roll back an entire unit of work, including uncommitted changes that are made from the calling application before the stored procedure call is made.

- Support for Windows Kerberos security lets you more easily manage workstation clients who seek access to data and services from heterogeneous environments.
- Global transaction support for distributed applications lets independent DB2 agents participate in a global transaction that is coordinated by an XA-compliant transaction manager on a workstation or a gateway server (Microsoft Transaction Server or Encina, for example).
- Support for a DB2 Connect Version 7 enhancement lets remote workstation clients quickly determine the amount of time that DB2 takes to process a request (the server elapsed time).
- Additional enhancements include:
  - Support for connection pooling and transaction pooling for IBM DB2 Connect
  - Support for DB2 Call Level Interface (DB2 CLI) bookmarks on DB2 UDB for UNIX, Windows, OS/2

---

## Features of DB2 for OS/390 and z/OS

Version 7 of DB2 UDB Server for OS/390 and z/OS offers several features that help you integrate, analyze, summarize, and share data across your enterprise:

- DB2 Warehouse Manager feature. The DB2 Warehouse Manager feature brings together the tools to build, manage, govern, and access DB2 for OS/390-based data warehouses. The DB2 Warehouse Manager feature uses proven technologies with new enhancements that are not available in previous releases, including:
  - DB2 Warehouse Center, which includes:
    - DB2 Universal Database Version 7 Release 1 Enterprise Edition
    - Warehouse agents for UNIX, Windows, and OS/390
    - Information Catalog
  - QMF Version 7
  - QMF High Performance Option
  - QMF for Windows
- DB2 Management Clients Package. The elements of the DB2 Management Clients Package are:
  - DB2 Control Center
  - DB2 Stored Procedure Builder
  - DB2 Installer
  - DB2 Visual Explain
  - DB2 Estimator
- Net Search Extender for in-memory text search for e-business applications
- Net.Data for secure Web applications

---

## Migration considerations

# Migration with full fallback protection is available when you have either DB2 for  
 # OS/390 Version 5 or Version 6 installed. You should ensure that you are fully  
 # operational on DB2 for OS/390 Version 5, or later, before migrating to DB2 for  
 # OS/390 and z/OS Version 7.

To learn about all of the migration considerations from Version 5 to Version 7, read the *DB2 Release Planning Guide* for Version 6 and Version 7; to learn about content information, also read appendixes A through F in both books.



---

## Chapter 2. Introduction to installation and migration

This chapter introduces you to the features and steps needed to install or migrate to DB2 Version 7.

*Installation* is the process of preparing DB2 to operate as an OS/390 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 and z/OS Version 7 from DB2 Version 6 or Version 5.** Whether you are installing or migrating, there are steps you must perform that are the same.

You must migrate to an OS/390 Version 2 Release 7 environment before installing DB2 Version 7.

Before you begin installing or migrating, plan the amount of direct access storage and virtual storage you need. “Chapter 3. Estimating DB2 storage needs” on page 25 helps you with your decisions. Planning and coordinating with other DB2 subsystems is essential if you plan to install the distributed data facility (DDF). For more information, see Part 2 (Volume 1) of *DB2 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 77 for help with your decisions.

DB2 Version 7 includes DB2 Installer, a workstation tool that is delivered as an element of the Management Tools Package. DB2 Installer enhances your productivity whether you are installing DB2 for OS/390 and z/OS for the first time or you are an experienced installer. DB2 Installer runs on your OS/2® or Windows NT® workstation. The DB2 Installer graphical interface illustrates the overall installation process and keeps a graphical record of how each subsystem is defined. For more information, see [www.ibm.com/software/db2os390/db2inst](http://www.ibm.com/software/db2os390/db2inst). You can define a basic DB2 subsystem quickly, or customize every installation option. Detailed online documentation is available in DB2 Installer and this document can be used for reference.

---

### Installing other features of the DB2 UDB Server for OS/390 and z/OS

For more information about installing the features of DB2 UDB for OS/390 and z/OS see the Program Directories for the features.

---

### Installation features

DB2 includes several features to help you perform the steps involved in installing or migrating to Version 7:

***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 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:

- Estimate storage needs
- Determine which new functions you will need
- If using distributed data, install VTAM and, optionally, TCP/IP
- Set up a System/390® 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 *DB2 ODBC Guide and Reference* for the additional installation jobs that you need to run.

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

- Install needed service on the prior release (if you are migrating)
- Check release incompatibilities and make changes in your applications
- 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 the version from which you migrated (Version 6 or Version 5)
- Remigrate to DB2 for OS/390 and z/OS Version 7

After you have completed migration, it is recommended that you avoid using new Version 7 facilities until you are certain that you will not need to fall back.

## Planning for migration incompatibilities

You should be aware that enhancements made to the release might cause unexpected results from your system utilities, applications or jobs. Proper planning should alleviate any system inconveniences. See “Make adjustments for release incompatibilities” on page 281 for more information.

## 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 1. Overview of SMP/E steps

Step	Description	Job
1	Copy and edit the SMP/E jobs	IEBCOPY
2	Initialize SMP/E environment for DB2 (optional)	DSNTIJAA
3	Allocate the libraries	DSNALLOC
4	Run the RECEIVE jobs	DSNRECV1, DSNRECV2, DSNRECV3, DSNRECV4
5	Run the clean-up job (optional)	DSNTIJUD
6	Run the APPLY jobs	DSNAPPL1, DSNAPPL2
7	Run the ACCEPT jobs	DSNACEP1, DSNACEP2
8	Unload the jobs for the additional FMIDs (optional)	
9	Run the receive jobs for the additional FMIDs (optional)	DSNRECV1, DSNRECV2, DSNRECV3, DSNRECV4
10	Run the apply jobs for the additional FMIDs (optional)	DSNAPPL1, DSNAPPL2
11	Run the accept jobs for the additional FMIDs (optional)	DSNACEP1, DSNACEP2
12	Copy and edit the SMP/E jobs for DB2 REXX Language Support (optional)	IEBCOPY
13	Run the RECEIVE job for DB2 REXX Language Support (optional)	DSNTTJRC
14	Run the APPLY job for DB2 REXX Language Support (optional)	DSNTTJAP
15	Run the ACCEPT job for DB2 REXX Language Support (optional)	DSNTTJAC

Table 1. Overview of SMP/E steps (continued)

Step	Description	Job
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 78 to run the installation CLIST, you can edit and run the jobs that install your DB2 Version 7 subsystem. The following steps install DB2 Version 7.

Table 2. Overview of steps for installing DB2 Version 7

Step	Description	Job
1	Define DB2 Version 7 to MVS, and build cataloged procedures	DSNTIJMV
2	Optionally, define a new integrated catalog facility catalog and 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	IPL MVS	—
12	Start DB2 Version 7	—
13	Define temporary work file table spaces, initial buffer pool sizes, and initial hiperpool sizes	DSNTIJTM
14	Define and bind DB2 objects and user-maintained databases	DSNTIJSG
15	Optionally, populate the user-maintained databases and if you are using DDF populate the communications database (within the DB2 catalog)	—
16	Bind the packages for DB2 REXX Language Support, if you have installed DB2 REXX Language Support.	DSNTIJRX
17	Image copy the DB2 catalog and directory	DSNTIJIC
18	Run the installation verification procedure	DSNTEJxx

For a detailed description of the installation procedure, see “Chapter 7. Installing the DB2 subsystem” on page 235.

## Migration steps summary

After you have performed the SMP/E steps and followed the steps on page 78 to run the installation CLIST, you can edit and run the jobs that migrate your Version 6 or Version 5 subsystem to a DB2 for OS/390 and z/OS Version 7 subsystem.

Migration to Version 7 includes the following steps:

*Table 3. Overview of steps for migrating to DB2 Version 7*

Step	Description	Job
1	Premigration activities to determine unsupported objects	DSNTIJPM
2	Optionally, 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 6/Version 5 subsystem.	—
3	Optionally, determine which plans and packages will be invalid after migration.	—
4	Optionally, check for consistency between catalog tables.	—
5	Image copy your Version 6/Version 5 catalog.	DSNTIJIC
6	Establish the DB2/TSO environment.	DSNTIJVC
7	Optionally, connect IMS to DB2.	—
8	Optionally, connect CICS to DB2.	—
9	Stop DB2 Version 6/Version 5.	—
10	Back up Version 6/Version 5 volumes.	—
11	Define DB2 initialization parameters.	DSNTIJUZ
12	Optionally, establish subsystem security.	—
13	Define DB2 Version 7 to MVS and build cataloged procedures.	DSNTIJMV
14	Define system data sets.	DSNTIJIN
15	Define authorization exit routines.	DSNTIJEX
16	IPL MVS	— <sup>1</sup>
17	Start DB2 Version 7	—
18	Tailor the DB2 Version 7 catalog	DSNTIJTC
19	Optionally, invoke the link checker (DSN1CHKR) to check for broken links on Version 7. Optionally, run job DSNTIJCX for checking indexes in the catalog and directory.	—
20	Prepare the dynamic SQL program and define initial buffer pool sizes and initial hiperspool sizes.	DSNTIJTM
21	Bind SPUFI and DCLGEN.	DSNTIJSG
22	Migrate SQL procedures tables.	DSNTIJMP
23	Bind the packages for DB2 REXX Language Support, if you have installed DB2 REXX Language Support.	DSNTIJRX
24	Image copy the Version 7 catalog.	DSNTIJIC
25	Run Version 6/Version 5 verification jobs Run Version 7 verification jobs	DSNTEJxx
24	Make adjustments for release incompatibilities.	—

**Note:**

<sup>1</sup> 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 the version from which you migrated (Version 6 or Version 5) after successfully completing the catalog migration to DB2 for OS/390 and z/OS Version 7. You can fallback if the catalog migration, job DSNTIJTC has completed successfully. You can only fallback to the version from which you migrated. You can fall back if a severe error occurs either during the subsequent migration steps or during operation of DB2 Version 7. **This process applies only to those who are migrating to Version 7.** It is described in 304 or 341. To fall back, perform the following steps:

Table 4. Overview of steps to fall back

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

---

## Remigration steps summary

Migration to DB2 for OS/390 and z/OS Version 7 after falling back to Version 6 or Version 5 (remigration) is simpler than migration. To remigrate to Version 7, perform the following steps:

Table 5. Overview of steps for remigration to DB2 Version 7

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 6/Version 5. Execute the queries in DSNTESQ to check for consistency between catalog tables.	DSN1CHKR
2	Image copy Version 6/Version 5	DSNTIJIC
3	Stop Version 6/Version 5	—
4	Reconnect TSO, IMS, and CICS to Version 7	—
5	Rename cataloged procedures	DSNTIJMV
6	Start DB2 Version 7	—
7	Optionally, image copy the Version 7 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 6/Version 5 verification jobs. Run Version 7 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 312 or “Remigrating” on page 347.





---

## Part 2. Planning and installing DB2

<b>Chapter 3. Estimating DB2 storage needs</b>	25
Disk storage for the DB2 subsystem	25
Disk requirements for the DB2 catalog	26
Disk requirements for the DB2 directory	26
Disk requirements for the active log data sets	27
Disk requirements for the bootstrap data sets	29
Disk requirements for the work file database	29
Disk requirements for the default database	32
Disk requirements for temporary table spaces	32
Determining the columns in the declared temporary table	32
Determining the lengths of the columns in the declared temporary table	33
Calculating the length of the longest declared temporary table row	33
Examples of determining the maximum row length for a declared temporary table	33
Disk requirements for the dump data set size	34
Disk requirements for the system databases	35
Disk requirements for the archive log data sets	35
Using the installation CLIST to calculate storage	35
Virtual storage for address spaces	36
DB2 distributed data facility address space (DSN1DIST)	36
IRLM address space (IRLMPROC)	36
DB2 system services address space (DSN1MSTR)	37
DB2 database services address space (DSN1DBM1)	37
Allied agent address space	37
Common service area	37
DB2-Established stored procedures address space (DSN1SPAS)	38
WLM-Established stored procedures address spaces	39
Virtual storage for storage pools and working storage	39
Buffer pool size calculation	40
Sort pool size calculation	41
RID pool size calculation	42
EDM pool size calculation	43
Calculating the EDM pool space needed for plans and packages	43
Calculating the EDM pool space for the prepared statement cache	45
Calculating the EDM pool data space for cached dynamic statements	46
Calculating the EDM pool space needed for database descriptors	46
Total EDM pool space	47
Data set control block storage size calculation	47
Working storage calculation	48
Virtual storage below the 16MB line	48
Real storage	49
<b>Chapter 4. Loading DB2 libraries</b>	51
What IBM sends you	51
What you produce	52
SMP/E step 1: Copy and edit the SMP/E jobs	55
Copying the SMP/E jobs	55
Editing the SMP/E jobs	56
Creating job statements	56
Choosing link list options	57
Accessing the correct DB2 program library	59
Performance considerations	59
DB2 library naming considerations	60

	SMP/E data set options . . . . .	61
	IRLM options . . . . .	62
	SMP/E step 2: Allocate the SMP/E CSI file and SMP/E control data sets:	
	DSNTIJAA (optional) . . . . .	62
	SMP/E step 3: Allocate distribution and target libraries: DSNALLOC . . . . .	64
	SMP/E step 4: Run the receive jobs: DSNRECV1, DSNRECV2, DSNRECV3,	
	DSNRECV4 . . . . .	64
	SMP/E step 5: Cleanup job for migration: DSNTIJUD (optional) . . . . .	65
	SMP/E step 6: Run the apply jobs: DSNAPPL1, DSNAPPL2 . . . . .	65
	SMP/E step 7: Run the accept jobs: DSNACEP1, DSNACEP2 . . . . .	66
I	SMP/E step 8: Unload the jobs for the additional FMIDs (optional) . . . . .	67
I	SMP/E step 9: Run the receive jobs for the additional FMIDs (optional) . . . . .	68
I	SMP/E step 10: Run the apply job for the additional FMIDs (optional). . . . .	68
I	SMP/E step 11: Run the accept job for the additional FMIDs (optional) . . . . .	68
	SMP/E step 12: Copy and edit the SMP/E jobs for DB2 REXX Language	
	Support (optional) . . . . .	68
	SMP/E step 13: Run the RECEIVE job for DB2 REXX Language Support	
	(optional) . . . . .	69
	SMP/E step 14: Run the APPLY job for DB2 REXX Language Support (optional) . . . . .	69
	SMP/E step 15: Run the ACCEPT job for DB2 REXX Language Support	
	(optional) . . . . .	69
	SMP/E step 16: Ensure installation of proper maintenance . . . . .	69
	Finishing SMP/E processing . . . . .	69
	<b>Chapter 5. Setting up and using DB2 Online Help . . . . .</b>	<b>71</b>
	Copying the bookshelf list, index, and books . . . . .	71
	Changing online help book data set names (optional). . . . .	71
	Customizing BookManager READ for online help (optional) . . . . .	72
	Verifying online help and setting exit options . . . . .	73
	Making online help accessible from installation and DB2I panels . . . . .	74
	Using online help . . . . .	74
	Accessing help with DB2 online help . . . . .	75
	Moving around in the DB2 online help . . . . .	75
	Searching for additional information . . . . .	75
	Exiting DB2 online help . . . . .	76
	<b>Chapter 6. Installing, migrating, and updating system parameters . . . . .</b>	<b>77</b>
	Running the installation CLIST . . . . .	77
	Making the DB2 ISPF libraries available to TSO. . . . .	77
	Invoking the CLIST . . . . .	78
	General instructions . . . . .	78
	Output from the panel session . . . . .	78
	Actions allowed on panels. . . . .	80
	Reading the panel descriptions . . . . .	80
	Directory of panels . . . . .	81
	Directory of panel field names . . . . .	82
	Online book data set names panel: DSNTIPA0 . . . . .	89
	Main panel: DSNTIPA1 . . . . .	90
	Recommended approach for a new installer . . . . .	95
	Data parameters panel: DSNTIPA2 . . . . .	96
	Define group or member panel: DSNTIPK . . . . .	101
	System resource data set names: DSNTIPH . . . . .	103
	Data set names panel 1: DSNTIPT . . . . .	107
	Data set names panel 2: DSNTIPU . . . . .	112
	Data set names panel 3: DSNTIPQ . . . . .	120
	Data set names panel 4: DSNTIPG . . . . .	123

Data set names panel 5: DSNTIPW . . . . .	127
Sizes panel 1: DSNTIPD . . . . .	130
Sizes panel 2: DSNTIP7 . . . . .	136
Thread management panel: DSNTIPE . . . . .	138
Buffer pool sizes panel 1: DSNTIP1 . . . . .	143
Buffer pool sizes panel 2: DSNTIP2 . . . . .	145
Buffer pool sizes panel 3: DSNTIP6 . . . . .	147
Tracing panel: DSNTIPN . . . . .	149
Operator functions panel: DSNTIPO . . . . .	153
Application programming defaults panel 1: DSNTIPF . . . . .	160
Application programming defaults panel 2: DSNTIP4 . . . . .	168
IRLM panel 1: DSNTIPI . . . . .	174
IRLM panel 2: DSNTIPJ . . . . .	179
Protection panel: DSNTIPP . . . . .	184
MVS parmlib updates panel: DSNTIPM . . . . .	188
Active log data set parameters: DSNTIPL . . . . .	192
Archive log data set parameters: DSNTIPA . . . . .	197
Databases and spaces to start automatically panel: DSNTIPS . . . . .	203
Distributed data facility: DSNTIPR . . . . .	205
Distributed data facility panel: DSNTIP5 . . . . .	210
Routine parameters panel: DSNTIPX . . . . .	214
Data definition control support panel: DSNTIPZ . . . . .	216
Job editing panel: DSNTIPY . . . . .	219
Install DB2 - CLIST calculations panel 1: DSNTIPC . . . . .	222
Install DB2 - CLIST calculations panel 2: DSNTIPC1 . . . . .	227
Completing the CLIST processing . . . . .	227
Responding to messages . . . . .	227
Tailoring the installation jobs . . . . .	228
Editing the subsystem parameters and DSNHDECP values . . . . .	231
The update process . . . . .	231
Updating parameters through the update selection menu panel: DSNTIPB . . . . .	231
Updating other parameters . . . . .	233
<b>Chapter 7. Installing the DB2 subsystem . . . . .</b>	<b>235</b>
Installation step 1: Define DB2 to MVS: DSNTIJMV . . . . .	235
DSNTIJMV updates to SYS1.PARMLIB . . . . .	236
DSNTIJMV updates to SYS1.PROCLIB . . . . .	239
Installation step 2: Define the integrated catalog facility catalog and Alias:	
DSNTIJCA . . . . .	240
Installation step 3: Define system data sets: DSNTIJIN . . . . .	240
Installation step 4: Initialize system data sets: DSNTIJID . . . . .	241
Installation step 5: Define DB2 initialization parameters: DSNTIJUZ . . . . .	242
Installation step 6: Define user exit routines: DSNTIJEX (optional) . . . . .	244
Installation step 7: Record DB2 data to SMF (optional). . . . .	244
Installation step 8: Establish subsystem security (optional) . . . . .	246
Installation Step 9: Connect DB2 to TSO . . . . .	246
Making DB2 load modules available to TSO and batch users . . . . .	246
Making DB2 CLISTs available to TSO and batch users: DSNTIJVC . . . . .	246
Ensuring that PL/I options are available . . . . .	247
Making panels, messages, and load modules available to ISPF and TSO . . . . .	247
Connecting DB2I panels to the ISPF main panel . . . . .	249
Establishing DB2 authorization IDs in TSO and RACF . . . . .	253
Installation step 10: Connect IMS to DB2 (optional) . . . . .	253
Installation step 11: Connect CICS to DB2 (optional) . . . . .	253
Installation step 12: IPL MVS . . . . .	253
Installation step 13: Start the DB2 subsystem . . . . .	254

	Installation step 14: Define temporary work files: DSNTIJTM. . . . .	255
	Installation Step 15: Define and bind DB2 objects and user-maintained databases: DSNTIJSG. . . . .	257
	Installation step 16: Populate the user-maintained databases (optional). . . . .	259
	Installation step 17: Bind the packages for DB2 REXX Language Support: DSNTIJRX (optional) . . . . .	259
	Installation Step 18: Back up the DB2 directory and catalog: DSNTIJIC . . . . .	259
	Authorizing DB2 users. . . . .	260
	Installation step 19: Verify the installation. . . . .	260
	Installing support for a communications network . . . . .	260
	Installing a second DB2 on the same operating system . . . . .	261
	Planning considerations . . . . .	261
	Installation procedure . . . . .	262
	Loading DB2 libraries . . . . .	262
	Tailoring installation jobs . . . . .	262
	Installing a second DB2 . . . . .	263
	Connecting attachment facilities . . . . .	264
	Preparing DB2 for use. . . . .	265
	Verifying your installation process . . . . .	266
	Considerations for using shared DASD . . . . .	266
#	Loading data with an SQL cursor. . . . .	266
	Enabling stored procedures after installation . . . . .	266
	Enabling DB2-supplied stored procedures . . . . .	268
	Enabling Control Center procedures . . . . .	268
	Enabling SQL procedure support. . . . .	269
#	Enabling CICS transaction invocation procedure . . . . .	269
	 <b>Chapter 8. Migrating from Version 5 to Version 7.</b> . . . . .	271
	Migration Considerations . . . . .	271
	Scrollable cursors are supported . . . . .	271
	Unicode support . . . . .	271
	Encoding schemes of string parameters for routines. . . . .	271
	Revoking SYSADM authorization. . . . .	272
	Changed messages from the LOAD utility . . . . .	272
	Change in recording soft errors in SYS1.LOGREC data set . . . . .	272
	More than 32 K databases are supported. . . . .	272
	Support for up to 150 000 connections. . . . .	272
	Log buffer size increased . . . . .	272
	Increased maximum number of data sets open . . . . .	272
	Changes to the RLST . . . . .	273
	Creating tables with DBCS and mixed columns . . . . .	273
	Increased size for generated code . . . . .	273
	Consider increasing IDBACK and CTHREAD . . . . .	273
	Customized DB2I defaults can be migrated . . . . .	273
	Migrating a data sharing group . . . . .	273
	Consider enlarging BSDS . . . . .	274
	Stored procedures . . . . .	274
	Utility enhancements . . . . .	275
	Work file database size calculations. . . . .	275
	Data set password protection is removed. . . . .	275
	SYSIBM.SYSPROCEDURES no longer used . . . . .	276
	Private protocol function not enhanced. . . . .	276
	Type 2 indexes are required . . . . .	276
	Shared read-only data is removed . . . . .	276
	DCE security authentication no longer supported . . . . .	276
	Release coexistence . . . . .	276

	IRLM . . . . .	276
	Data sharing . . . . .	277
	Distributed environment . . . . .	277
I	Migration step 1: Premigration activities . . . . .	278
	Remove incomplete table definitions . . . . .	278
	Identify unsupported objects (DSNTIJPM) . . . . .	279
	Convert indexes to type 2 . . . . .	279
	Remove data set passwords . . . . .	280
	Eliminate shared read-only data . . . . .	280
	Remove views on two catalog tables . . . . .	280
	Save critical access paths (optional) . . . . .	280
	DRDA support for three-part names (optional) . . . . .	281
	Examine all new and changed values for DB2I panels . . . . .	281
	Make adjustments for release incompatibilities . . . . .	281
	Adjust application programs. . . . .	281
	Fewer sort operations for queries. . . . .	284
	Sysplex query parallelism in the PLAN_TABLE. . . . .	285
	Limit backouts with system restarts . . . . .	285
	DISPLAY BUFFERPOOL changes . . . . .	285
	Index changes . . . . .	285
	ALTER INDEX syntax . . . . .	285
	RECOVER INDEX becomes REBUILD INDEX. . . . .	286
	Work space formulas changed for utilities . . . . .	286
#	Default for FASTSWITCH parameter is YES . . . . .	286
	Change to parameter in IRLMPROC startup procedure. . . . .	286
	Migration Step 2: Run link checker on DB2 for OS/390 Version 5 table spaces (optional). . . . .	286
	Migration step 3: Determine which plans and packages are invalid after migration (optional) . . . . .	287
	Migration step 4: Check for consistency between catalog tables (optional)	288
	Migration step 5: Image copy directory and catalog in case of fallback: DSNTIJIC . . . . .	288
	Migration step 6: Connect DB2 to TSO . . . . .	289
	Making DB2 load modules available to TSO and batch users . . . . .	290
	Making DB2 CLISTs available to TSO and batch users: DSNTIJVC . . . . .	290
	Making panels, messages, and load modules available to ISPF and TSO . . . . .	291
	Migration step 7: Connect IMS to DB2 (optional) . . . . .	292
	Migration step 8: Connect CICS to DB2 (optional) . . . . .	292
	Migration step 9: Stop DB2 Version 5 activity . . . . .	292
	Migration step 10: Back up your DB2 for OS/390 Version 5 volumes (optional)	293
	Migration step 11: Define DB2 initialization parameters: DSNTIJUZ . . . . .	293
	DSNTIJUZ actions . . . . .	294
	Additional steps . . . . .	294
	Considerations for the BSDS . . . . .	294
	Migration step 12: Establish subsystem security (optional) . . . . .	295
	Migration step 13: Define DB2 Version 7 to MVS: DSNTIJMV . . . . .	295
	DSNTIJMV actions . . . . .	295
	Completing the step . . . . .	297
	Migration step 14: Define system data sets: DSNTIJIN. . . . .	297
	Migration step 15: Define user authorization exit routines: DSNTIJEX . . . . .	297
	Migration step 16: IPL MVS. . . . .	298
	Migration step 17: Start DB2 for OS/390 and z/OS Version 7 . . . . .	299
	Migration step 18: Tailor DB2 for OS/390 and z/OS Version 7 catalog: DSNTIJTC . . . . .	300
	Migration step 19: Ensure there are no problems with the catalog (optional)	301
	Migration step 20: Prepare dynamic SQL program: DSNTIJTM. . . . .	301

Migration step 21: Bind SPUFI and DCLGEN and user maintained database activity: DSNTIJSG . . . . .	301
Migration step 22: Migrate SQL procedures tables: DSNTIJMP (optional) . . . . .	302
Migration step 23: Bind the packages for DB2 REXX Language Support: DSNTIJRX (optional) . . . . .	302
Migration step 24: Image copy DB2 for OS/390 and z/OS Version 7 catalog: DSNTIJIC . . . . .	302
Migration step 25: Verify your DB2 for OS/390 and z/OS Version 7 system . . . . .	303
Running the DB2 Version 5 sample jobs . . . . .	303
Running the Version 7 sample jobs . . . . .	303
Testing Version 7 with your application test cases . . . . .	304
Falling back . . . . .	304
Fallback considerations . . . . .	304
Data sharing . . . . .	304
Frozen objects . . . . .	304
Automatic rebind . . . . .	306
Other fallback considerations . . . . .	306
Fallback procedure . . . . .	308
Fallback Step 1: Run phase 0 of the DB2 for OS/390 and z/OS Version 7 installation verification procedure . . . . .	309
Fallback Step 2: Stop DB2 for OS/390 and z/OS Version 7 activity . . . . .	309
Fallback Step 3: Reactivate DB2 Version 5 code: DSNTIJFV . . . . .	310
Fallback Step 4: Reconnect TSO, IMS, and CICS to DB2 Version 5 . . . . .	310
Fallback Step 5: Start DB2 Version 5 . . . . .	310
Fallback Step 6: Verify fallback . . . . .	312
Remigrating . . . . .	312

## **Chapter 9. Migrating from Version 6 to Version 7. . . . . 315**

Migration Considerations . . . . .	315
Scrollable cursors are supported . . . . .	315
Unicode support . . . . .	315
New default encoding scheme . . . . .	316
Revoking SYSADM authorization . . . . .	316
Encoding schemes of string parameters for routines . . . . .	316
Changed messages from the LOAD utility . . . . .	316
Creating tables with DBCS and mixed columns . . . . .	316
Increased size for generated code . . . . .	316
Consider increasing IDBACK and CTHREAD . . . . .	316
Change in recording soft errors in SYS1.LOGREC data set . . . . .	316
Customized DB2I defaults can be migrated . . . . .	317
Migrating a data sharing group . . . . .	317
Work file database size calculations . . . . .	317
SYSIBM.SYSPSM and SYSIBM.SYSPSMOPTS no longer used . . . . .	318
Release coexistence . . . . .	318
IRLM . . . . .	318
Data sharing . . . . .	318
Distributed environment . . . . .	319
Migration step 1: Premigration activities . . . . .	319
Remove incomplete table definitions . . . . .	319
Save critical access paths (optional) . . . . .	320
Examine all new and changed values for DB2I panels . . . . .	321
DRDA support for three-part names (optional) . . . . .	321
Make adjustments for release incompatibilities . . . . .	321
Adjust application programs . . . . .	321
DCE security authentication no longer supported . . . . .	323
Fewer sort operations for queries . . . . .	323

#

Sysplex query parallelism in the PLAN_TABLE. . . . .	323
DISPLAY BUFFERPOOL changes . . . . .	323
Index changes . . . . .	323
Work space formulas changed for utilities . . . . .	323
Default for FASTSWITCH parameter is YES . . . . .	324
Migration Step 2: Run link checker on DB2 for OS/390 Version 6 table spaces (optional). . . . .	324
Migration step 3: Determine which plans and packages are invalid after migration (optional) . . . . .	325
Migration step 4: Check for consistency between catalog tables (optional)	325
Migration step 5: Image copy directory and catalog in case of fallback: DSNTIJIC . . . . .	326
Migration step 6: Connect DB2 to TSO . . . . .	327
Making DB2 load modules available to TSO and batch users . . . . .	327
Making DB2 CLISTs available to TSO and batch users: DSNTIJVC . . . . .	327
Making panels, messages, and load modules available to ISPF and TSO	328
Migration step 7: Connect IMS to DB2 (optional) . . . . .	329
Migration step 8: Connect CICS to DB2 (optional) . . . . .	329
Migration step 9: Stop DB2 Version 6 activity . . . . .	330
Migration step 10: Back up your DB2 for OS/390 Version 6 volumes (optional)	331
Migration step 11: Define DB2 initialization parameters: DSNTIJUZ . . . . .	331
DSNTIJUZ actions . . . . .	331
Additional steps . . . . .	331
Considerations for the BSDS . . . . .	332
Migration step 12: Establish subsystem security (optional) . . . . .	332
Migration step 13: Define DB2 Version 7 to MVS: DSNTIJMV . . . . .	332
DSNTIJMV actions . . . . .	333
Completing the step . . . . .	334
Migration step 14: Define system data sets: DSNTIJIN. . . . .	335
Migration step 15: Define user authorization exit routines: DSNTIJEX . . . . .	335
Migration step 16: IPL MVS. . . . .	336
Migration step 17: Start DB2 for OS/390 and z/OS Version 7 . . . . .	336
Migration step 18: Tailor DB2 for OS/390 and z/OS Version 7 catalog: DSNTIJTC . . . . .	337
Migration step 19: Ensure there are no problems with the catalog (optional)	338
Migration step 20: Prepare dynamic SQL program: DSNTIJTM. . . . .	338
Migration step 21: Bind SPUFI and DCLGEN and user maintained database activity: DSNTIJSG . . . . .	339
Migration step 22: Migrate SQL procedures tables: DSNTIJMP (optional)	340
Migration step 23: Bind the packages for DB2 REXX Language Support: DSNTIJRX (optional) . . . . .	340
Migration step 24: Image copy DB2 for OS/390 and z/OS Version 7 catalog: DSNTIJIC . . . . .	340
Migration step 25: Verify your DB2 for OS/390 and z/OS Version 7 system	340
Running the DB2 Version 6 sample jobs . . . . .	340
Running the Version 7 sample jobs . . . . .	341
Testing Version 7 with your application test cases . . . . .	341
Falling back . . . . .	341
Fallback considerations . . . . .	342
Data sharing . . . . .	342
Frozen objects . . . . .	342
Automatic rebind. . . . .	343
Other fallback considerations . . . . .	343
Fallback procedure . . . . .	344
Fallback Step 1: Run phase 0 of the DB2 for OS/390 and z/OS Version 7 installation verification procedure . . . . .	344

I

Fallback Step 2: Stop DB2 for OS/390 and z/OS Version 7 activity . . . . .	344
Fallback Step 3: Reactivate DB2 Version 6 code: DSNTIJFV . . . . .	345
Fallback Step 4: Reconnect TSO, IMS, and CICS to DB2 Version 6 . . . . .	345
Fallback Step 5: Start DB2 Version 6 . . . . .	346
Fallback Step 6: Verify fallback . . . . .	347
Remigrating . . . . .	347
<b>Chapter 10. Verifying with the sample applications . . . . .</b>	<b>349</b>
Installation verification phases and programs . . . . .	350
Planning for verification . . . . .	354
Special considerations for COBOL programs . . . . .	355
Special considerations for C and C++ programs . . . . .	357
Special considerations for PL/I programs . . . . .	358
Phase 0: Deleting the sample objects (DSNTEJ0) . . . . .	359
Phase 1: Creating and loading sample tables . . . . .	360
Job DSNTEJ1 . . . . .	360
Job DSNTEJ1L . . . . .	362
Job DSNTEJ1P . . . . .	363
Job DSNTEJ1U . . . . .	364
Phase 2: Testing the batch environment . . . . .	364
Job DSNTEJ2A . . . . .	364
Job DSNTEJ2C . . . . .	365
Job DSNTEJ2D . . . . .	365
Job DSNTEJ2E . . . . .	366
Job DSNTEJ2F . . . . .	366
Job DSNTEJ2P . . . . .	367
Job DSNTEJ2U . . . . .	367
Phase 3: Testing SPUFI, DRDA Access, dynamic SQL, and TSO . . . . .	369
Testing SPUFI . . . . .	370
Running SPUFI at remote non-DB2 systems . . . . .	370
Running dynamic SQL and the ISPF/CAF application . . . . .	371
Jobs DSNTEJ3C and DSNTEJ3P: . . . . .	371
Starting an application in an ISPF/TSO environment. . . . .	373
Phase 4: Testing the IMS environment. . . . .	373
Jobs DSNTEJ4C and DSNTEJ4P . . . . .	374
Starting an application in an IMS environment . . . . .	376
Using the phone application in IMS . . . . .	377
Phase 5: Testing the CICS environment . . . . .	377
Job DSNTEJ5A . . . . .	377
Jobs DSNTEJ5C and DSNTEJ5P . . . . .	377
Starting an application in a CICS environment . . . . .	380
Using the phone application in CICS . . . . .	380
Using CICS storage-handling facilities . . . . .	381
Phase 6: Accessing data at a remote site. . . . .	381
DRDA Access sample . . . . .	382
Job DSNTEJ6. . . . .	382
DB2 Private Protocol Access sample . . . . .	383
Stored procedure samples . . . . .	384
Stored procedure sample without result set . . . . .	384
Job DSNTEJ6S . . . . .	385
Job DSNTEJ6P . . . . .	385
Stored procedure sample with result set . . . . .	386
Job DSNTEJ6T . . . . .	386
Job DSNTEJ6D . . . . .	386
Sample utilities stored procedure. . . . .	387
Job DSNTEJ6U . . . . .	387



I	Job DSNTSJ6V . . . . .	388
I	Job DSNTSJ6W . . . . .	388
I	Job DSNTSJ6Z . . . . .	389
	Sample ODBA stored procedure . . . . .	389
	Job DSNTSJ61 . . . . .	390
	Job DSNTSJ62 . . . . .	390
	Sample SQL procedures . . . . .	390
	Job DSNTSJ63 . . . . .	391
	Job DSNTSJ64 . . . . .	391
	Job DSNTSJ65 . . . . .	392
	Starting an application in an ISPF/TSO environment in phase 6 . . . . .	392
	Phase 7: Accessing LOB Data. . . . .	393
	Job DSNTSJ7 . . . . .	393
	Job DSNTSJ71 . . . . .	394
	Job DSNTSJ73 . . . . .	394
	Job DSNTSJ75 . . . . .	395
	Starting an application in an ISPF/TSO environment in phase 7 . . . . .	395
	The sample applications . . . . .	395
	Printing the sample application listings. . . . .	396
	Specifying values in the sample application panels . . . . .	396
	Allowable combinations . . . . .	398
	Specifying data values . . . . .	398
	Function keys . . . . .	398
	Scenarios . . . . .	399
	The project application scenario . . . . .	399
	Updating an activity. . . . .	400
	The organization application scenario . . . . .	401
	Starting a new operation . . . . .	403
	Adding a new department . . . . .	403
	Deleting an entry. . . . .	403
	Transferring . . . . .	404
	The phone application scenario . . . . .	405
	Phone application panels. . . . .	406
	Using the phone application under batch . . . . .	407
	The distributed organization application scenario . . . . .	409
	Displaying department structure at the local location. . . . .	409
	Displaying department information at the local location. . . . .	410
	Updating a department at the local location . . . . .	411
	Adding an employee at a remote location. . . . .	414
	Erasing an employee at a remote location . . . . .	415
	Employee resume and photo scenarios . . . . .	417
	Sample LOB table . . . . .	418
	LOB application panels for Resume scenario . . . . .	418
	LOB application panels for Photo scenario . . . . .	420
	Edit exit routine . . . . .	420
	Huffman compression exit routine . . . . .	421
	Sample field procedure . . . . .	421
	Dynamic SQL statements (DSNTESA, DSNTESQ) . . . . .	421
	DSNTESA . . . . .	421
	DSNTESQ . . . . .	423
	Dynamic SQL programs (DSNTIAD, DSNTJ2, DSNTJAL) . . . . .	423
	<b>Chapter 11. Connecting the IMS attachment facility. . . . .</b>	<b>425</b>
	Making DB2 load modules available to IMS . . . . .	425
	Defining DB2 to IMS . . . . .	425
	Defining new programs and transactions to IMS . . . . .	429

Defining DB2 plans for IMS applications (optional) . . . . .	429
Generating a user language interface (optional) . . . . .	430
IMS attachment facility macro (DSNMAPN) . . . . .	431
Description . . . . .	431
<b>Chapter 12. Connecting the CICS attachment facility</b> . . . . .	433
Using the attachment facility for CICS Version 4 and CICS TS Version 1.1 . . . . .	433
Calculating space requirements for the CICS attachment facility . . . . .	433
CICS Version 4 . . . . .	434
Estimating storage needed for CICS attachment threads . . . . .	435
Defining the DB2 to CICS connection using the RCT . . . . .	436
Updating the CICS system tables . . . . .	436
System initialization table entries . . . . .	436
Transaction entries . . . . .	437
Program entries . . . . .	439
PLT processing . . . . .	440
Providing CICS support for DB2 commands . . . . .	441
Updating CICS initialization JCL . . . . .	442
Coordinating DB2 and CICS security . . . . .	442
Preparing CICS applications for DB2 . . . . .	443
CICS attachment facility macro (DSNCRCT) . . . . .	444
DSNCRCT TYPE=INIT . . . . .	444
Description . . . . .	445
DSNCRCT TYPE=COMD . . . . .	451
Description . . . . .	451
DSNCRCT TYPE=POOL . . . . .	452
Description . . . . .	452
DSNCRCT TYPE=ENTRY . . . . .	453
Description . . . . .	454
DSNCRCT TYPE=FINAL . . . . .	461
Description . . . . .	461
DSNCRCT TYPE=DSECT . . . . .	461
Description . . . . .	461

---

## Chapter 3. Estimating DB2 storage needs

This chapter describes the following aspects of storage:

- “Disk storage for the DB2 subsystem”
- “Virtual storage for address spaces” on page 36
- “Virtual storage for storage pools and working storage” on page 39
- “Real storage” on page 49.

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 disk storage beyond what this estimate. 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. For specific estimates, refer to the *DB2 Program Directory*. 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, 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 *DB2 Data Sharing: Planning and Administration*.

---

### Disk storage for the DB2 subsystem

Refer to the *DB2 Program Directory* for tables showing estimated space requirements for specific environments.

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 “Disk requirements for the active log data sets” on page 27. 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 6 on page 26. 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 6. 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 7 provides estimated 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 disk 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.

*Table 7. 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

For information about disk requirements for DB2 libraries and SMP/E data sets, see the *DB2 Program Directory*.

## Disk requirements for the DB2 catalog

Storage requirements for the entire set of DB2 catalog data sets and their indexes are shown in Table 8.

*Table 8. 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 Part 5 (Volume 2) of *DB2 Administration Guide*.

## Disk 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 9 on page 27.

Table 9. 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

## Disk 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 disk 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 four 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 disk 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 a LOB table space is defined with LOG(YES), you should estimate 1.0 to 1.1 times the size of the LOB for INSERTs or UPDATES. Only control information is logged for DELETes. With LOG(NO), only internal control information is logged. The formula is:

$$\text{MAX (50 bytes, 0.1 * (size of LOB))}$$

If you enabled data sharing, you generally need to have more DASD for the active log and archive the logs more frequently. See Chapter 5 of *DB2 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. However, using large log data sets causes DB2 to archive less frequently. Infrequent archiving can result in increased data loss in the event of remote site recovery or disaster recovery. You can avoid some outages by having an adequate number of active logs for several hours of batch update processing.

Table 10 shows estimated storage requirements for active log data sets assuming dual logging. Table 11 shows the amount of space required for active log data sets on various IBM DASD devices. The estimates in both of these tables include DASD track overhead.

Table 10. 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	36000	392	2352
Extra Large	24	72000	784	4704

Table 11. 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 a media type described in Table 12 on page 29, the planning size numbers 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, the tape utilization is fairly low.

Table 12. 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 the estimated planning sizes 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 the estimated planning sizes 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 DSNTIPL specifies either the number of consecutive log records written between DB2 system checkpoints or the number of minutes per checkpoint. 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 1 million, then the time needed to restart DB2 after a termination without a quiesce can grow to over 30 minutes. The recommended values for the LOGLOAD range are in the range of 25 000 to 1 million. The recommended values for the checkpoint frequency are in the range of 25 000 to 500 000 in log records or 2 to 5 in minutes.

- 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 *DB2 Administration Guide*.

## Disk requirements for the bootstrap data sets

Each bootstrap data set (BSDS) requires 1MB for each data set. If you are installing, DB2 automatically allocates two copies of the BSDS. If you are migrating, Version 7 adopts the BSDS characteristics you specified for your previous version. That is, if you specified two copies of the BSDS data sets for your previous version, you will have two copies for Version 7. Keeping two copies of the BSDS data set is strongly recommended. The total space requirement is about 2MB for both BSDSs. The BSDSs at any size site require about 4 cylinders, regardless of device type.

## Disk requirements for the work file database

The work file database is used as storage for processing SQL statements that require working storage. Table 13 on page 30 shows the disk requirement estimates for the temporary work files in the work file database. For additional migration

considerations when running DSNTIJTC, see “Chapter 8. Migrating from Version 5 to Version 7” on page 271 or “Chapter 9. Migrating from Version 6 to Version 7” on page 315 .

*Table 13. Estimated space requirements (in cylinders) for the work file database by site size*

<b>Site Size</b>	<b>3380</b>	<b>3390</b>	<b>9340</b>
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:

*Records per page* = MIN(MAXROWS, FLOOR (4076 / (*data* + *key* + *prefix*))),  
but cannot exceed 255 (the value of MAXROWS)

*Total pages* = CEILING (*rows* / *Records per page*)

*Total segments* = CEILING (*Total pages* / 24)

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.

When a large object (LOB ) column is part of a result table, and the result table must be placed in a work file for sorting, the actual LOB column data is not placed



in the work file. Therefore, LOB columns do not require large increases in the amount of work file space that DB2 requires. For work file calculations, users should allow 51 bytes of storage per LOB column needed by the work file.

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 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)
- $\text{Rows} = 45327$
- $\text{Records per page} = \text{MIN}(\text{MAXROWS}, \text{FLOOR} (4076 / (36 + 36 + 6))) = 52$
- $\text{Total pages} = \text{CEILING} (45327 / 52) = 872$
- $\text{Segments} = \text{CEILING} (872 / 24) = 37$
- $\text{Tracks} = \text{CEILING} (872 / 12) \times 1.5 = 111$

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:

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

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 Part 5 (Volume 2) of *DB2 Administration Guide*.

## Disk 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 the default database size.

## Disk requirements for temporary table spaces

DB2 uses declared temporary tables for processing scrollable cursors. Therefore, before you can use a scrollable cursor, your database administrator needs to create a TEMP database and TEMP table spaces for those declared temporary tables. For example:

```
CREATE
DATABASE DTTDB AS TEMP;
CREATE TABLESPACE DTTTS IN DTTDB
  SEGSIZE 4;
```

If there is more than one TEMP table space in the subsystem, DB2 chooses the table spaces to use for scrollable cursors.

The page size of the TEMP table space must be large enough to hold the longest row in the declared temporary table. The size of a row in the declared temporary table might be considerably larger than the size of the row in the table for which the scrollable cursor is used. The size of the row depends on these factors:

- The number of columns that are stored in the declared temporary table
- The size of each column

The number of columns in the declared temporary table depends on these factors:

- The number of columns in the select list of the SELECT statement for the cursor
- The number of expressions in the select list that contain more than a single column name
- If the SELECT statement contains an ORDER BY clause, the number of columns in the ORDER BY clause
- Whether the result table is read-only

To calculate the size of the longest row in the declared temporary table, follow these steps:

1. Determine the columns in the declared temporary table. See “Determining the columns in the declared temporary table”.
2. Determine the length of each column in the declared temporary table. See “Determining the lengths of the columns in the declared temporary table” on page 33.
3. Calculate the total length of the longest declared temporary table row. See “Calculating the length of the longest declared temporary table row” on page 33.

### Determining the columns in the declared temporary table

The columns that are in the declared temporary table for a scrollable cursor depend on whether the result table of the cursor is read-only.

For a read-only result table, the following items are columns in the declared temporary table:

- Each expression in the select list
- Each column in the ORDER BY clause that is not in the select list
- One additional column that is generated by DB2

For a result table that is not read-only, the following items are columns in the declared temporary table:

- Each column in the select list
- Each expression in the select list that contains more than a single column name
- Three additional columns that are generated by DB2

### **Determining the lengths of the columns in the declared temporary table**

After you determine the columns that are in the declared temporary table, you need to determine the length of each column. Use the following method:

1. For columns other than the columns that DB2 generates, determine the data type of each column. See "Data types of result columns" table, which is in Chapter 4 of *DB2 SQL Reference*, for this information.
2. Determine the length of each column, based on the data type, in the following way:
  - For a declared temporary table column that is the result of the concatenation of CHAR, VARCHAR, GRAPHIC or VARGRAPHIC data types, use the numbers in the "Data type and length of concatenated operands" table, which is in Chapter 2 of *DB2 SQL Reference*. Add one byte if the column is nullable.
  - For a declared temporary table column that is the result of an expression that contains LOB columns, specify a length of 120 for each column, literal or host variable that is referenced in the expression. Then add the byte counts for the expression.
  - For a LOB data type, specify a length of 120.
  - For a ROWID data type, specify a length of 42.
  - For a declared temporary table column of any other data type, use the "Byte counts of columns by data type" table, which is in Chapter 5 of *DB2 SQL Reference*, to determine the column length. Add one byte if the column is nullable.
3. For the columns that DB2 generates, determine the total length for those columns in the following way:
  - If the result table of cursor is read-only, the length for the added column is 22 bytes.
  - If the result table of cursor is not read-only, the total length for the three added columns is 33 bytes.

### **Calculating the length of the longest declared temporary table row**

To determine the length of the longest declared temporary table row, add the column lengths that you calculated in the previous step.

### **Examples of determining the maximum row length for a declared temporary table**

Suppose that table T1 has the following columns:

Column name	Data type
C1	CLOB(100M)
C2	CHAR(10)
C3	VARCHAR(100) NOT NULL
C4	INTEGER NOT NULL
C5	INTEGER

Now suppose that you declare two scrollable cursors for the table:

```
DECLARE CUR1 INSENSITIVE SCROLL CURSOR FOR
  SELECT C1, C2 || C3, C4 FROM T1
  WHERE C3 > :HV;

DECLARE CUR2 SENSITIVE STATIC SCROLL
CURSOR FOR
  SELECT C1, C2 || C3, C4 FROM T1
  WHERE C3 > :HV;
```

The result table for cursor CUR1 is read-only. Therefore, the columns, column lengths, and maximum row length of the declared temporary table for CUR1 are:

Column	Data type	Effective length
C1	CLOB(100M)	120
C2    C3	VARCHAR(110)	113
C4	INTEGER NOT NULL	4
C5	INTEGER	5
One column added by DB2		22
Total length		264

The result table for cursor CUR2 is not read-only. Therefore, the columns, column lengths, and maximum row length of the declared temporary table for CUR2 are:

Column	Data type	Effective length
C1	CLOB(100M)	120
C2	CHAR(10)	11
C3	VARCHAR(100) NOT NULL	102
C4	INTEGER NOT NULL	4
C2    C3	VARCHAR(110)	113
Three columns added by DB2		33
Total length		383

## Disk requirements for the dump data set size

**Recommendation:** Use these guidelines for the dump data sets:

- At least 2 dump data sets
- Approximately 200 cylinders of 3390 disk 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 *OS/390 MVS 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.

## Disk 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.

## Disk 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 “Disk requirements for the active log data sets” on page 27 and “Disk requirements for the bootstrap data sets” on page 29 to determine these sizes, or see the messages generated by the CLIST on page 222. 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 may 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 disk storage estimates in the installation CLIST. **Recommendation:** Use the model site estimates the first time you install DB2. Use the model approach described in “Disk storage for the DB2 subsystem” on page 25 to estimate DB2 disk use. After your site has some experience in operating DB2, you can recalculate your disk 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 installation CLISTs and REXX EXEC. The DSNTCALC EXEC contains most of the data set calculations. You can

run the CLIST to calculate the sizes. Installation panel DSNTIPC on page 222 displays the storage sizes calculated by the CLIST.

---

## Virtual storage for address spaces

DB2 uses these types of private address spaces:

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

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.

**Recommendation:** 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 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. Use the command `MODIFY irImproc,STATUS,STOR` to monitor IRLM storage. Adjust the installation parameter values for IRLM after you gain some experience with DB2.

You can adjust the MAXCSA parameter dynamically using the `MODIFY irImproc,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 *DB2 Command Reference*.

## DB2 system services address space (DSN1MSTR)

The DB2 system services address space needs less space than the database services address space. **Recommendation:** 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 39.

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 433.

## 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 approximately calculated by these guidelines:

- Up to 40KB for each DB2 subsystem
- Add 24KB for each IRLM started
- Add 1KB for every 13 latch contentions
- Add 4KB for every 4 notify requests.

To estimate storage that 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
- Start with 73KB for IRLM
- Add 1.9MB for IRLM non-optional trace buffers
- Add 1.9MB for IRLM optional trace buffers

- 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

If you use the distributed data functions of DB2, you will probably find that you need more virtual storage. You can expect your storage needs to increase in the extended common storage area (ECSA) above the 16MB line by:

- 1 KB for each conversation, plus
- 2 KB for each thread that uses distributed processing, plus
- 1 KB for each DB2 site in your network, plus
- 40 KB for code that relates to distributed processing

Your virtual storage needs under normal conditions will probably not increase in the CSA below the 16MB line. In a data sharing environment, if an IRLM performs member recovery or structure rebuild additional virtual storage is required. Any other increase in the amount of virtual storage needed occurs within the extended private area of the DB2 database address space and the extended private area of the distributed data address space.

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.

**Recommendation:** Use 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 Part 6 of *DB2 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 and z/OS 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 Part 5 (Volume 2) of *DB2 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. **Recommendation:** 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 Section 1 of *DB2 Administration Guide*.

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 Part 5 (Volume 2) of *DB2 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 222.

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. Optionally, virtual buffer pools can be allocated in data spaces. If data spaces are used for some or all buffer pools, then the storage demands for the DBM1 address space decreases. 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. A value of 60KB or more for improved performance is recommended. 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, ten buffer pools for 8KB page sets, ten buffer pools for 16KB page sets, and ten buffer pools for 32KB table spaces. The ALTER BUFFERPOOL command can make the changes dynamically while DB2 is up. See Part 5 (Volume 2) of *DB2 Administration Guide* for information on changing the buffer pool sizes.

Use Table 14 to calculate the virtual buffer pool sizes for your subsystem.

Table 14. Virtual buffer pool size calculation

Virtual Buffer Pool Calculation		Default	
Buffers for BP0	_____ x 4KB = _____	2000 x 4KB	= 8000KB
Buffers for BP1	+ _____ x 4KB = _____	+ 0 x 4KB	= 0KB
Buffers for BP2	+ _____ x 4KB = _____	+ 0 x 4KB	= 0KB
.			
.			
.			
Buffers for BP49	+ _____ x 4KB = _____	+ 0 x 4KB	= 0KB

Table 14. Virtual buffer pool size calculation (continued)

Virtual Buffer Pool Calculation		Default	
Buffers for BP8K0	+ ____ x 8KB = ____	+ 0 x 8KB	= 0KB
Buffers for BP8K1	+ ____ x 8KB = ____	+ 0 x 8KB	= 0KB
.			
Buffers for BP16K0	+ ____ x 16KB = ____	+ 0 x 16KB	= 0KB
Buffers for BP16K1	+ ____ x 16KB = ____	+ 0 x 16KB	= 0KB
.			
Buffers for BP32K	+ ____ x 32KB = ____ = ____	+24 x 32KB	= 768KB = 8768KB
.			
Buffers for BP32K9	+ ____ x 32KB = ____ = ____	+ 0 x 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 under "Buffer Pools in Part 1 (Volume 1) of *DB2 Administration Guide*.

Hiperpool sizes for each virtual buffer pool are specified during installation. If you do not specify a size, the hiperpool is not created. A hiperpool cannot exist unless it is associated with a virtual buffer pool. See "Buffer pool sizes panel 1: DSNTIP1" on page 143, "Buffer pool sizes panel 2: DSNTIP2" on page 145, and "Buffer pool sizes panel 3: DSNTIP6" on page 147 for details about the installation options for hiperpool sizes. Your hiperpool sizes are not committed during installation. You can change the size at 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 hyperspaces. Hiperpools fully use a 2GB hyperspace 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 operation:

Minimum sort storage = 240KB

Maximum sort storage = 64MB

DB2 will initially allocate 240KB for each sort and will gradually add more storage until the maximum sort work area limit is reached or the maximum 16K nodes are

populated at the bottom of the sort tree, whichever occurs first. With 16K nodes at the bottom of the sort tree, the average run size for each sorted string is 32K records.

The default size of the sort pool is 1MB, but you can override this default value by entering the desired sort pool 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 Part 5 (Volume 2) of *DB2 Administration Guide* for instructions on choosing sizes for optimal performance.

**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 always 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 the disk 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 that is allocated to complete sort processing, you must allocate more disk space for the work file database. For more information, see “Disk requirements for the work file database” on page 29.

## 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

The default size of the RID pool is 4MB, but you can override this default value by entering the desired RID pool size on installation panel DSNTIPC. If you do this, estimate the storage required for the RID pool with the following formula:

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

where *A* is:

- 4 for non-large tables
- 5 for large tables

See Part 5 (Volume 2) of *DB2 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, packages, database descriptors, and cached prepared statements. These estimations do not account for every factor that affects EDM pool size. Your actual EDM pool size may vary. Generally, it is recommended 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 222.

### 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:

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

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 plan during execution is typically 12KB.

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 querying the SYSPLAN and SYSPACKAGE catalog tables. PLSIZE or PKSIZE is the size of the base segment of the plan or package. AVGSIZE is the average size for each 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,
CASE WHEN MAX(SECTNOI) <> 0
THEN MAX(SECTNOI)
ELSE MAX(SECTNO)
END AS SECTNUM
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,
CASE WHEN MAX(SECTNOI) <> 0
THEN MAX(SECTNOI)
ELSE MAX(SECTNO)
END AS SECTNUM
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 2.2KB (1.8KB + 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 data space for cached dynamic statements

To relieve storage constraints in the DBM1 address space, you can choose to have the part of your EDM pool that contains cached dynamic statements in a data space. Data spaces are only used when you specify YES for CACHE DYNAMIC SQL on installation panel DSNTIP4.

DB2 determines the default size of the data space as follows:

- If the EDM pool storage size is less than or equal to 40 MB, the EDM data space is 40 MB.
- If the EDM pool storage size is larger than 40 MB, the EDM data space size equals the EDM pool size.

You can override this value on installation panel DSNTIPC.

## 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 15, 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 15. 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:

- Let *concpplans* be the number of concurrently executing plans. This is the sum of the values specified as MAX USERS and MAX REMOTE ACTIVE fields 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 DSNTIPE.
- Let *dbdsize* be the DBD size.

Then, calculate as follows:

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

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  $((70 + 64) + 50) \times 25\text{KB} + (100 \times 72\text{KB})$ , or 11800KB. Then, 50KB are added overhead, and the total of 11850KB is multiplied by 1.25 to give the result of 14812KB.

## 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:

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

You can modify DSMAX by editing job DSNTIJUZ. The maximum number of concurrently open data sets is typically:

- 20 000 to 25 000 if you are running on OS/390 Version 2 Release 6 or later
- 10 000 if you are running on a level of OS/390 that is earlier than Version 2 Release 6

See Part 5 (Volume 2) of *DB2 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 3600KB  $\times$  1.8KB, or 5400KB.

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 Part 5 (Volume 2) of DB2 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:

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

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

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

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 Part 5 (Volume 2) of *DB2 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 Section 1 of *DB2 Administration Guide* 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 600KB + 70 + 64 + (3000 × 0.3KB), or 1634KB. 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 16 shows total main storage calculation. Place your estimates in the spaces provided and make the indicated calculations.

Table 16. Main storage size calculation

Category	Your Size	Default
EDM pool storage size		14812KB
Buffer pool size	+	+8768KB
Sort pool size	+	+1000KB
RID pool size	+	+4000KB
Data set control block storage size	+	+5400KB
Code storage size	+4300KB	+4300KB
Working storage size	+	+5960KB
Total main storage size (above 16MB line)	=	=44240KB
Region Size (below 16MB line) (assume SWA above the line)		1634KB

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 222.

## 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 17 on page 50 and Table 18 on page 50.

Table 17. Real storage size calculation

Category	Default	Virtual Size	× Factor	= Real Size	See Page
Buffer pools	8768KB	_____	1.0	_____	Note <sup>1</sup>
Sort pool	1000KB	_____	0.5	_____	41
RID pool	4000KB	_____	0.5 +	_____	42
EDM pool	14812KB	_____	1.0 +	_____	43
Data set size	5400KB	_____	0.6 +	_____	47
Code size + 1100 (4 address spaces)	5400KB	5400KB	0.5 +	2700KB	
Working storage + (DSCF + DDF)	6960KB	_____	1.0 +	_____	48
Log buffers	400KB	_____	0.8 +	_____	Note <sup>1</sup>
Lock space	5000KB	_____	0.4 +	_____	Note <sup>1</sup>
CSA/ECSA	2160KB	_____	0.4 +	_____	37
Total default real storage size =					= _____

**Note:** <sup>1</sup> See Part 5 (Volume 2) of *DB2 Administration Guide*

Table 18. Default real storage size calculation

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 19 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 19. Additional real storage for more users

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

---

## Chapter 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 7**, 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 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 6. Installing, migrating, and updating system parameters” on page 77.

---

### What IBM sends you

When you order DB2, you receive standard label 9-track magnetic tapes recorded at 6250 BPI, 3480 cartridges, or 4mm cartridges, depending on the feature you ordered. If you order a custom built product delivery offering (CBPDO), your order may differ.

DB2 Management Clients Package, Net.Data, and DB2 REXX Language Support are separately ordered as non-priced features of DB2.

The DB2 Management Clients Package includes: DB2 Installer, Estimator, Visual Explain, DB2 Connect, and DB2 for OS/390 and z/OS Control Center Enablement. Most of the tools are offered to you on a CD-ROM. For more information on these workstation features see their readme files on the CDs. The DB2 for OS/390 and z/OS Control Center Enablement is shipped on a tape or cartridge. Directions for installing the Control Center Enablement are in the DB2 Management Clients Package Program Directory. For customization information for the Control Center, see *DB2 Connect Enterprise Edition for OS/2 and Windows: Quick Beginnings* and [www-4.ibm.com/software/data/db2/os390/cc390](http://www-4.ibm.com/software/data/db2/os390/cc390).

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 *DB2 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.

Along with these tapes or cartridges, you receive a set of documents. One of these documents is *IBM DATABASE 2 Universal Database Server for OS/390 and z/OS Program Directory*. Read the *DB2 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 a Program Directory for each feature of the DB2 server that you ordered. Read these directories before installing any of the DB2 features.

If you plan to use DB2's callable SQL interface (DB2 ODBC), there are additional installation jobs that you need to run. See *DB2 ODBC Guide and Reference* for more information.

If you plan to use DB2 for OS/390 and z/OS Java Edition, see *DB2 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 *DB2 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 is loaded into 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 20 and Table 21 on page 53 describe all the DB2 distribution and target libraries. The distribution libraries contain the master copy of all elements for your DB2 system.

The storage requirements for target and distribution libraries are listed in the *DB2 Program Directory*.

*Table 20. DB2 distribution libraries*

Distribution Libraries	Description
<i>prefix.ADSNBASE</i>	This library contains all jobs that are required to complete SMP/E installation.
<i>prefix.ADSNLOAD</i>	This library contains an individual object module for every DB2 load module. It contains the IRLM load modules if you choose to install IRLM into the same distribution libraries as DB2.
<i>prefix.ADSNLOD2</i>	This library contains a PDS/E dataset, which houses JDBC and SQLJ DLLs.

Table 20. DB2 distribution libraries (continued)

Distribution Libraries	Description
<i>prefix</i> .ADSNMACS	This library contains the DB2 macros, sample programs, sample data, initialization data, TSO CLISTS, ISPF panels, and ISPF messages.
<i>prefix</i> .ADSNENU or ADSNDKF	This library contains the DB2 English or Kanji task panels, respectively.
<i>prefix</i> .ADSNIVPD	This library contains the IVP input data and expected output for sample applications.
<i>prefix</i> .ADXRLoad	This library contains an individual object module for every IRLM load module.
<i>prefix</i> .ADXRSAMP	This library contains the installation procedures for installing IRLM Version 2.
<i>prefix</i> .ADSNHFS	This library contains the data to be copied into the MVS/Open Edition HFS.
<i>prefix</i> .ADSNBKS <i>prefix</i> .ADSNINDX <i>prefix</i> .ADSNINST <i>prefix</i> .ADSNSHLF	

Table 21. DB2 target libraries

Target Libraries	Description
<i>prefix</i> .SDSNBASE	This library contains all jobs that are required to complete SMP/E installation.
<i>prefix</i> .SDSNBKS	This library contains the BookManager® books that are used for DB2 online help. See “Chapter 5. Setting up and using DB2 Online Help” on page 71 for more information.
<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 7.
<i>prefix</i> .SDSNEXIT	This program library is empty when first created. The installation jobs put DSNHDECP, the DSNZPxxx subsystem parameters load module, and user exit modules into this library.
<i>prefix</i> .SDSNINDX	This library contains the BookManager index for the books in <i>prefix</i> .SDSNINST. See “Chapter 5. Setting up and using DB2 Online Help” on page 71 for more information.
<i>prefix</i> .SDSNINST	This library contains jobs that are used to install DB2 online help. See “Chapter 5. Setting up and using DB2 Online Help” on page 71 for more information.
<i>prefix</i> .SDSNLINK	This library contains ERLY code of Version 7.
<i>prefix</i> .SDSNLOAD	This library contains Version 7 load modules.
<i>prefix</i> .SDSNLOD2	This library contains the PDS/E dataset, which houses JDBC and SQLJ DLLs.

Table 21. DB2 target libraries (continued)

Target Libraries	Description
<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> .SDSNSHLF	This library contains the BookManager bookshelf for the books in <i>prefix</i> .SDSNINST. See “Chapter 5. Setting up and using DB2 Online Help” on page 71 for more information.
<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 routing 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> .SDSNC.H	This contains the header files. CLI requires header files. C language application programs can use header files.
<i>prefix</i> .SDSNIVPD	This contains the IVP input data and the expected output for sample applications.
<i>prefix</i> .SDXRSAMP	IRLM samples library. This library may be empty if you chose to install IRLM elsewhere.
<i>prefix</i> .SDXRRESL	This library contains the IRLM load modules. This library may be empty if you chose to install IRLM elsewhere.

The remainder of this chapter explains how to edit and run 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 22.

Table 22. List of SMP/E jobs

Job Name	Description
DSNTIJAA	This job creates the DB2 target and distribution zones, and defines the SMP/E control data sets in these zones and in the SMP/E global zone.
DSNACEP1	This job invokes SMP/E to accept all the required FMIDs into the DB2 distribution libraries (DLIBs).
DSNACEP2	This job invokes SMP/E to accept all the additional FMIDs into the DB2 distribution libraries (DLIBs).



Table 22. List of SMP/E jobs (continued)

Job Name	Description
DSNALLOC	This is the SMP/E <i>allocation job</i> . It creates the DB2 target and distribution libraries and defines the libraries in the SMP/E target and distribution zones for DB2 for the required and additional FMIDs.
DSNAPPL1	This job invokes SMP/E to apply all the FMIDs to the DB2 target libraries for the required FMIDs.
DSNAPPL2	This job invokes SMP/E to apply all the FMIDs to the DB2 target libraries for the additional FMIDs.
DSNRECV1	This job invokes SMP/E to receive all the required FMIDs (from both tapes or cartridges) from the base tape into the SMP/E control data sets.
DSNRECV2	This job invokes SMP/E to receive all the required FMIDs (from both tapes or cartridges) for IRLM into the SMP/E control data sets.
DSNRECV3	This job invokes SMP/E to receive all the additional FMIDs (from both tapes or cartridges) for ODBC, JDBC, and SQLJ into the SMP/E control data sets.
DSNRECV4	This job invokes SMP/E to receive FMIDs (from both tape or cartridge) 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 Version 6 or Version 5 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 1 on page 56. Check the Program Directory which is shipped with the product for any changes to the contents of the tapes or cartridges.

If you have a CBPDO or ServerPac, 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 IEBPTPCH to print each job. If you need additional information about these utilities, see *DFSMS/MVS: Utilities*.

```

//STEP1      EXEC PGM=IEBCOPY
//SYSPRINT   DD SYSOUT=*
//IN         DD DSN=IBM.HDB7710.F5,UNIT=tunit, VOL=SER=DB7710,
//           LABEL=(6,SL),DISP=(OLD,KEEP)
//OUT        DD DSN=jcl-library-name,
//           DISP=(NEW,CATLG,DELETE),
//           VOL=SER=dasdvol,UNIT=SYSALLDA
//           DCB=*,STEP1.IN,SPACE=(TRK,(20,10,10))
//SYSUT3     DD UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSIN      DD *
            COPY INDD=IN,OUTDD=OUT
/*

```

Figure 1. Sample JCL to copy SMP/E jobs to DASD

Change the lowercase parameters in the above sample to uppercase values that meet your site's requirements before submitting, where *tunit* is the unit value matching the product tape or cartridge, *jcl-library-name* is the name of the data set where the sample jobs will reside, and *dasdvol* is the volume serial of the DASD device where the data set will reside.

If this job fails or abends, correct the job and rerun it.

This JCL copies all members required to complete SMP/E installation. These jobs currently exist as members of the partitioned data set IBM.HDB7710.F5 on tape VOLSER=DB7710.

After running the copy job, edit and run jobs DSNALLOC, DSNTIJUD (optional), DSNRECV1, DSNRECV2, DSNRECV3, DSNRECV4, DSNAPPL1, DSNACEP1, DSNAPPL2, and DSNACEP2. See the header notes within each job for information on how to customize the job for your particular installation. Do not run DSNRECV2 if your IRLM is at a higher level. For more information, see the *DB2 Program Directory*.

## 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	56
Link List Options	57
DB2 Program Library	59
DB2 Library Naming Considerations	60
SMP/E Data Set Options	61
IRLM Options	62

### 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.SDSNBASE:** Contains all jobs that are required to complete SMP/E installation.

**prefix.SDSNLINK:** Contains modules that you must place in the link list because they are loaded at subsystem initialization during IPL. For Version 7, the load module library SDSNLINK contains modules that are called early (ERLY) code. If your system is at the prerequisite maintenance level, your Version 6 early code is *upward* compatible with Version 7. The Version 7 early code is *downward* compatible with Version 6 and Version 5.

**If you are migrating,** be aware that any maintenance to early 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.SDSNLOAD2*: Contains a PDS/E data set, which houses JDBC and SQLJ DLLs.

***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 LNKST 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:*** The IRLM load module DXRRL183 must be in the link list if the MVS TRACE CT command (COMP = irlmssnm) is issued for the IRLM subsystem name. 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 DSNALLOC 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 DSNALLOC.
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 7 subsystems ). If this is the case, read the suggestions on page 58. In addition, create separate libraries for DSNHDECP and user-written exits of each DB2 subsystem. For considerations in data sharing environments, see *DB2 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 7 production subsystem and a DB2 Version 7 test subsystem at different service levels. Or, you can have two DB2 subsystems at different release levels. For instance, you can have a Version 7 subsystem and a Version 6 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 DSNALLOC 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 7 already contain the necessary JOBLIB or STEPLIB statements. In addition, the startup procedures that DB2 provides for a previous version and Version 7 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 442 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 LNKSTxx. The TSO attach modules are DSNECP00, DSNECP10, DSNECM00, 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 LNKSTxx.
- 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, DSNECM00, 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 20 on page 52. 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 7 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:

<i>prefix</i> .ADSNLOAD	<i>prefix</i> .SDSNMACS
<i>prefix</i> .SDSNCLST	<i>prefix</i> .SDSNSAMP
<i>prefix</i> .SDSNEXIT	<i>prefix</i> .DBRMLIB.DATA
<i>prefix</i> .SDSNLINK	<i>prefix</i> .RUNLIB.LOAD
<i>prefix</i> .SDSNLOAD	<i>prefix</i> .SRCLIB.DATA
<i>prefix</i> .SDSNDBRM	<i>prefix</i> .SDSNIVPD
<i>prefix</i> .SDXRRESL	<i>prefix</i> .SDSNC.H

**Recommendation:** 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 62.

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, DSNALLOC, 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 (JCI4106) 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 *OS/390 SMP/E User's Guide*.

**Establishing SMP/E data sets for two releases:** A single set of SMP/E zone structures can record only one release of DB2. We strongly recommend that you maintain separate zone structures for both a previous version (Version 5 or Version 6) and Version 7 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 (DSNALLOC), it creates a set of SMP/E data sets. If you choose to reuse your previous version's zone structure, you can run job DSNTIJUD to delete SMP/E data for the previous version. 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. **Recommendation:** Copy 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 the SMP/E CSI file and SMP/E control data sets: DSNTIJAA (optional)

This job creates the SMP/E consolidate software inventory (CSI) file and SMP/E data sets and allocates them to SMP/E. DSNTIJAA also creates the DB2 target and distribution zones. DSNTIJAA is required only if these objects are not created and allocated to the SMP/E global zone. 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.



For each group of data sets, DSNTIJAA **requires** a data set prefix and a volume name on which to allocate the data sets. These names are called the *allocation job parameters*.

Change the allocation parameters according to the decisions you made regarding the SMP/E data set options. Table 23 lists the allocation job parameters (prefix and volume) for each of the three groups of data sets.

Table 23. Allocation job parameters for SMP/E control data sets

Parameter Type	Search Strings
Prefix	?SMPPRE?
TLIB Prefix	?TLIBPRE?
Volume	?SMPVOL?
TLIB Volume	?TLIBVOL?
Middle-level Qualifier	?SMPMLQ?

If you are using JES3, you must split job DSNALLOC 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 DSNTIJAA. 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 35 MB (1 MB=1048576 B) of free space. That is about 37 cylinders on a 3390 and 49 cylinders on a 3390.

The block size is specified as 19069 for unformatted data sets and is determined by the system for other data sets. **Important:** AVGREC requires that SMS be active however the data sets don't have to be allocated on SMS-managed storage devices.

The space allocations in DSNTIJAA 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 (400,400,1200).  
| This is a minimum; change it only to increase it.  
|

**SMP/E zone structure:** SMP/E zone structures are discussed in the *OS/390 SMP/E User's Guide*. You can choose to use a different zone structure from the one shown in DSNTIJAA.

---

## SMP/E step 3: Allocate distribution and target libraries: DSNALLOC

This job creates the DB2 target and distribution libraries and defines them in the SMP/E target and distribution zones for DB2. DSNALLOC also creates the target and distribution libraries for DB2 Online Help (FMID HDB771A). If you do not want DB2 Online Help, you can delete the JCL in DSNALLOC that creates ADSNBKS, ADSNINDEX, ADSNINST, ADSNSHLF, SDSNBKS, SDSNINDEX, SDSNINST, and SDSNSHLF libraries.

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, DSNALLOC **requires** a data set prefix and a volume name on which to allocate the data sets. These names are called the *allocation job parameters*.

You need to change the allocation parameters according to the decisions you made regarding the LNKST option and library definition. Table 24 lists the allocation job parameters (prefix, volume, and unit name) for each of the three groups of data sets.

Table 24. Allocation job parameters for DB2 distribution and target libraries

Data Sets	Parameter Type	Search Strings
DB2 Distribution Libraries	Prefix	?DLIBPRE?
	Volume	?DLIBVOL?
DB2 Target Libraries	Prefix	?TARGPRE?
	Volume	?TARGVOL?

---

## SMP/E step 4: Run the receive jobs: DSNRECV1, DSNRECV2, DSNRECV3, DSNRECV4

Before you run the next three jobs, create backups of your previous version's 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 jobs load the DB2 program modules, macros, and procedures, IRLM, ODBC, JDBC, SQLJ, and Kanji 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.

Use the IRLM (FMID HIR2101), that is shipped as part of DB2 Version 7. If the IRLM already installed on your system is at a higher maintenance level than the IRLM shipped with DB2 Version 7, you can remove the HIR2101 step from job DSNRECV2. If you use the same level IRLM code for your IMS and DB2

| subsystems, use separate SMP/E zones or SMP/E control data sets. Using  
| down-level IRLM code increases your IRLM service activity and is not  
| recommended.

| If you do not want DB2 Online Help, remove FMID HDB771A from the list of FMIDs  
| to be received by DSNRECV1.

---

## SMP/E step 5: Cleanup job for migration: DSNTIJUD (optional)

**Recommendation:** Avoid running this job by using new SMP/E zones for your migration. If this is not possible (you are installing Version 7 in the same SMP/E libraries in which you installed Version 6 or Version 5), you must run job DSNTIJUD.

Job DSNTIJUD is run for migration from a previous version to ensure that delete processing is done properly before installing Version 7. It performs necessary SMP/E cleanup by deleting all entries from the previous version 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.

Use the IRLM (FMID HIR2101), which is shipped as part of DB2 Version 7, unless you have a higher maintenance level of IRLM already installed on your system. If you want to use different levels of IRLM for your IMS and DB2 subsystems, 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 statement from this job before executing:

- DEL MOD(DFHEAI)
- DEL MOD(DFHEAIO)
- DEL MOD(ISPLINK)

Job DSNTIJUD should be run before the SMP/E APPLY (jobs DSNAPPL1 and DSNAPPL2). 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 6: Run the apply jobs: DSNAPPL1, DSNAPPL2

| The SMP/E APPLY jobs, DSNAPPL1 and DSNAPPL2, copy and link-edit the DB2  
| program modules, macros, and procedures into the DB2 target libraries for required  
| FMIDs and additional FMIDs, respectively..

Examine the jobs before you run them. 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 the APPLY job.

If you do not apply the FMIDs in a single APPLY statement as DSNAPPLx does, use the following order:

1. HIY7710, HIZ7710, and HBD7710 together
2. HIR2101.

Expect a return code of 4 from this job. You may receive warning messages GIM23913W, GIM61903W, IEW2454W, and IEW2646W during execution of DSNAPPL1. 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 you plan to use the DB2 Call Level Interface (CLI), see *DB2 ODBC Guide and Reference* for information on the CLI FMID.

If the IRLM (FMID HIR2101) on your system is a more current release or has had maintenance applied so that it is the same or higher maintenance level than the one shipped with DB2, remove FMID HIR2101 from the APPLY step of job DSNAPPL1.

If you do not want an element, remove the element's FMID from the list of FMIDs to be applied by DSNAPPL2.

Table 25. Product FMIDs

Product Name	FMID
Text Extender	JDB771C
IAV Extenders	JDB661B
XML Extender	JDB771X
ODBC	JDB7717
JDBC/SQLJ	JDB7712
Online Help	HDB771A

---

## SMP/E step 7: Run the accept jobs: DSNACEP1, DSNACEP2

The SMP/E ACCEPT jobs, DSNACEP1 and DSNACEP2, copies the program modules, macros, and procedures into the DB2 distribution libraries for the required FMIDs and additional FMIDs, respectively. 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 DSNACEPx.

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 the accept jobs

If this job fails or abends, correct the problem and rerun the job.

If the IRLM (FMID HIR2101) on your system is a more current release than the one shipped with DB2 or has had maintenance applied so that it is the same or higher level of maintenance, remove FMID HIR2101 from the ACCEPT step of job DSNACEPx.

If you do not want an element, remove the element's FMID from the list of FMIDs to be accepted by DSNACEPx.

Table 26. Product FMIDs

Product Name	FMID
Text Extender	JDB771C
IAV Extenders	JDB771B
XML Extender	JDB771X
ODBC	JDB7717
JDBC/SQLJ	JDB7712
Online Help	HDB771A

## SMP/E step 8: Unload the jobs for the additional FMIDs (optional)

This step is optional unless you plan to use the additional FMIDs. The additional FMIDs are:

Table 27. Product FMIDs

Product Name	FMID
Online Help books and bookshelves	HDB771A
DB2 Kanji panels	JDB7712
JDBC/SQLJ	JDB7712
ODBC	JDB7717
IAV Extenders	JDB771B
Text Extender	JDB771C
XML Extender	JDB771X

For each additional FMID use the sample JCL in Figure 2 as a guide for copying the jobs from the tape. Make sure that you specify the correct VOLSER, UNIT information, and data set names.

```
//STEP1 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//IN DD DSN=IBM.HDB7710.F5,UNIT=tunit,VOL=SER=DB7710),
// LABEL=(6,SL),DISP=(OLD,KEEP)
//OUT DD DSN=jcl-library-name,
// DISP=(NEW,CATLG,DELETE),
// VOL=SER=dasdvol,UNIT=SYSALLDA
// DCB=*.STEP1.IN,SPACE=(TRK,(20,10,10))
//SYSUT3 DD UNIT=SYSALLDA,SPACE=(CYL,(1,1))
//SYSIN DD *
COPY INDD=IN,OUTDD=OUT
/*
```

Figure 2. Sample JCL for copying SMP/E jobs to disk

where *tunit* is the unit value matching the product tape or cartridge, *jcl-library-name* is the name of the data set where the sample jobs will reside, and *dasdvol* is the volume serial of the disk device where the data set will reside.

---

## SMP/E step 9: Run the receive jobs for the additional FMIDs (optional)

This step is optional unless you plan to use any of the additional FMIDs. To use the additional FMIDs, run the corresponding receive job that is listed in Table 28. Use “SMP/E step 4: Run the receive jobs: DSNRECV1, DSNRECV2, DSNRECV3, DSNRECV4” on page 64 as a guide to help you with this job.

Table 28. Product FMIDs

Product Name	FMID	Job Name
Online Help books and bookshelves	HDB771A	DSNRECV1
DB2 Kanji panels	JDB7711	DSNRECV4
JDBC/SQLJ	JDB7712	DSNRECV3
ODBC	JDB7717	DSNRECV3
IAV Extenders	JDB661B	DMBRECEV
Text Extender	JDB771C	DESRECEV
XML Extender	JDB771X	DXXRECEV

---

## SMP/E step 10: Run the apply job for the additional FMIDs (optional)

This step is optional unless you plan to use the additional FMIDs. To use the additional FMIDs, change job DSNAPPL2 to remove any unwanted FMIDs. Use “SMP/E step 6: Run the apply jobs: DSNAPPL1, DSNAPPL2” on page 65 as a guide to help you with this job.

---

## SMP/E step 11: Run the accept job for the additional FMIDs (optional)

This step is optional unless you plan to use the additional FMIDs. To use the additional FMIDs, change job DSNACEP2 to remove any unwanted FMIDs. Use “SMP/E step 7: Run the accept jobs: DSNACEP1, DSNACEP2” on page 66 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 (optional)

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 3 on page 69 to invoke the MVS utility IEBCOPY to copy the SMP/E jobs to DASD.

```

/* COMPID: DB2,5740XYR00
/* DOC:   LOAD REXX SMP INSTALLATION JCL FROM TAPE FOR DB2
//LOAD    EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//JCLTAPE DD DSN=IBM.JDB771H.F1,VOL=(PRIVATE,,SER=DB771H),
//          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 3. Sample JCL to Copy 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 (optional)**

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 (optional)**

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 (optional)**

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 Version 6 or Version 5 subsystem must be at the proper maintenance level *before* migrating to Version 7. Refer to *DB2 Program Directory* for information on that proper maintenance level.

---

## **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 ISPPLIB to select the current display language. Following is an example of a logon procedure:

```

/*                                THIS VERSION DISPLAYS ENGLISH PANELS */
//ISPPLIB DD DSN=prefix.SDSNSPFP,DISP=SHR      ENGLISH TASK
//          DD DSN=prefix.SDSNPFPE,DISP=SHR      ENGLISH DB2I

/*                                THIS VERSION DISPLAYS JAPANESE PANELS */
//ISPPLIB 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 ISPPLIB. Following is an example of a CLIST:

```

/* Execute this CLIST outside of ISPF */
PROC 0 LANGUAGE(E)
FREE DD(ISPPLIB)
WRITE DO YOU WANT ENGLISH OR JAPANESE PANELS: Enter E or J.
READ &LANGUAGE;
IF &LANGUAGE = E +
    THEN ALLOC DD(ISPPLIB) DS('DSN710.SDSNSPFP' 'DSN710.SDSNPFPE') +
        SHR /*ENGLISH*/
    ELSE ALLOC DD(ISPPLIB) DS('DSN710.SDSNSPFP' 'DSN710.SDSNPFPK') +
        SHR /*JAPANESE*/
END

```

Some users allocate the ISPF panel library from their DEFAULT CLIST. Allocation of ddname ISPPLIB controls the current language just as it does for the LOGON procedure.

**If you are falling back to a previous version,** change your logon procedures or CLISTs that use the Kanji feature to point to the previous version's libraries.



---

## Chapter 5. Setting up and using DB2 Online Help

The online help for DB2 for OS/390 and z/OS Version 7 lets you select information in an online DB2 book.

To use online help, complete the setup instructions in this chapter and then invoke the DSNTINS0 CLIST.

If you choose not to use online help, invoke the DSNTINST CLIST as described in “Invoking the CLIST” on page 78.

Before you install online help, you must first install BookManager READ. See *BookManager READ/MVS V1R3: Installation Planning & Customization* for information on how to do this.

After you have installed BookManager READ, see the following topics for the information you need to install, customize, and use online help:

- Copying the bookshelf list, index, and books
- Changing online help book data set names (optional)
- Customizing BookManager READ for online help (optional)
- Verifying online help and setting exit options
- Making online help accessible from installation and DB2I panels
- Using online help

---

### Copying the bookshelf list, index, and books

The bookshelf list, index, and books are shipped as partitioned data sets, but you must copy the contents to sequential data sets before they can be used. Submit the sample JCL DSNUNL1 to copy the bookshelf list to a sequential data set. Submit the sample JCL DSNUNL2 to copy the bookshelf, index, and books to sequential data sets. DSNUNL1 and DSNUNL2 are in data set DSN710.SDSNINST.

---

### Changing online help book data set names (optional)

Copies of the following DB2 books are shipped with online help:

- *DB2 Application Programming and SQL Guide*
- *DB2 Command Reference*
- *DB2 Installation Guide*
- *DB2 Utility Guide and Reference*

The default names for those books are DSNHELP.*book-ID*.BOOK. You can change the high-order qualifier of the book data set names, but if you do, you must indicate to DB2 what the new data set names are. Do this when you start DB2 installation or DB2I by changing the data set names on installation panel DSNTIPA0 or DB2I panel DSNEIPA0.

To access these panels, do the following:

- For installation, start the installation procedure by running CLIST DSNTINS0. See “Chapter 6. Installing, migrating, and updating system parameters” on page 77 for more information.

- For DB2I, specify YES in field CHANGE HELP BOOK NAMES? in the DB2I Defaults panel. See Part 5 of *DB2 Application Programming and SQL Guide* for more information.

After you enter the data set names in panel DSNTIPA0 or DSNEIPA0, DB2 saves them so that you need to modify the panel again only if you change the data set names. For example, if you want DB2 Online Help to access the latest versions of the DB2 books, you can copy those books from the DB2 for OS/390 and z/OS Version 7 Licensed Online Library CD-ROM to MVS sequential data sets, then access DSNTIPA0 or DSNEIPA0 and enter the new book data set names.

---

## Customizing BookManager READ for online help (optional)

You can perform the following actions to make BookManager READ and DB2 Online Help work better together:

- Add the DB2 Online Help library to the BookManager library list

This lets you select DB2 Online Help 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= 'DSNSHnnn.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.
- 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      */
```

- 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 DSNSHnnn 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.DSNShnnn.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.DSNShnnn.BKSHELF'

F1=Help F12=Cancel
```

5. The following message appears:

```
The bookshelf data set 'DSNHELP.DSNShnnn.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
```

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
__ DSNShnnn DB2 V6 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.

**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.

---

## Making online help accessible from installation and DB2I panels

To make online help accessible from the installation and DB2I task panels, concatenate the 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 ISPTLIB is allocated.

---

## Using online help

DB2 Online Help links you directly from DB2 panels to the information you need in the DB2 online books. Because DB2 Online Help uses BookManager READ to display the help text, you can perform any BookManager function after online help opens a DB2 book. This section explains how to access online help from installation or DB2I panels and how to navigate within the 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. DB2 Online Help starts and opens a DB2 online book to the topic.

## Moving around in the DB2 online help

DB2 Online Help lets you 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<sup>®</sup> 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, online help 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. Online help 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 DB2 Online Help, 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.

---

## Chapter 6. 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 a previous version to DB2 for OS/390 and z/OS Version 7. In between, you can consider **updating** many of them at any time to improve your operations.

When installing or migrating, you can run the installation CLIST or DB2 Installer 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 5. Setting up and using DB2 Online Help” on page 71.
- 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 *OS/390 MVS Initialization and Tuning Guide*.

### 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 JCL edited with the values you entered 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

Figure 4 on page 79 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 246 for more information. If you are migrating, see “Making DB2 CLISTs available to TSO and batch users: DSNTIJVC” on page 290.



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.

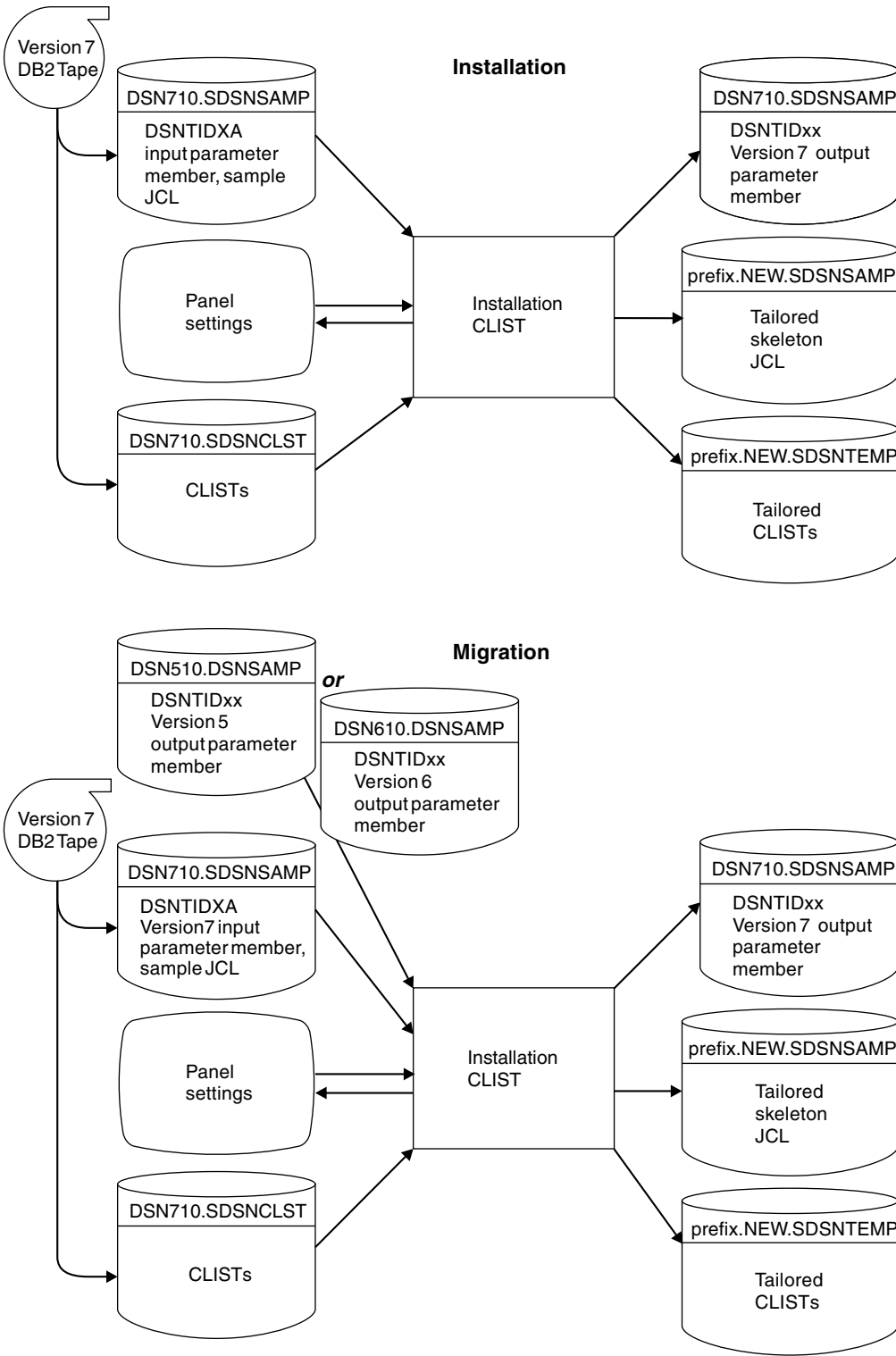


Figure 4. Examples of input to and output from the installation CLIST

## 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 that you are now using as input, 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 DSNTIDxx data set and are not reflected on the panels.

**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 the value associated with that field**. If the maximum allowable value is over 1024, in most cases you can use the equivalent K value. (The CLIST automatically multiplies the K value by 1024.) If the maximum allowable value is over 1 048 576, in most cases you can use 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 the update panel 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 29. Panel identifiers

Panel ID	Panel Title	Page
DSNTIPA0	Online Book Data Set Names	89
DSNTIPA1	Main Panel	90
DSNTIPA2	Data Parameters	96
DSNTIPK	Define Group or Member	101
DSNTIPH	System Resource Data Set Names	103
DSNTIPT	Data Set Names Panel 1	107
DSNTIPU	Data Set Names Panel 2	112
DSNTIPQ	Data Set Names Panel 3	120
DSNTIPG	Data Set Names Panel 4	123
DSNTIPW	Data Set Names Panel 5	127
DSNTIPD	Sizes Panel 1	130
DSNTIP7	Sizes Panel 2	136
DSNTIPE	Thread Management	138
DSNTIP1	Buffer Pool Sizes Panel 1	143
DSNTIP2	Buffer Pool Sizes Panel 2	145
DSNTIP6	Buffer Pool Sizes Panel 3	147
DSNTIPN	Tracing and Checkpoint Parameters	149
DSNTIPO	Operator Functions	153
DSNTIPF	Application Programming Defaults Panel 1	160
DSNTIP4	Application Programming Defaults Panel 2	168
DSNTIPI	IRLM Panel 1	174
DSNTIPJ	IRLM Panel 2	179
DSNTIPP	Protection	184
DSNTIPM	MVS PARMLIB Updates	188
DSNTIPL	Active Log Data Set Parameters	192
DSNTIPA	Archive Log Data Set Parameters	197
DSNTIPS	Databases and Spaces to Start Automatically	203
DSNTIPR	Distributed Data Facility Panel 1	205
DSNTIP5	Distributed Data Facility Panel 2	210
DSNTIPX	Routine Parameters	214
DSNTIPZ	Data Definition Control Support	216
DSNTIPY	Job Editing	219

Table 29. Panel identifiers (continued)

Panel ID	Panel Title	Page
DSNTIPC	CLIST Calculations Panel 1	222
DSNTIPC1	CLIST Calculations Panel 2	227
DSNTIPB	Update Selection Menu	231

## Directory of panel field names

Table 30. Panel fields

Panel Field Name	Panel	Page
ALLOCATION UNITS	DSNTIPA	197
APPL PROG AND SQL GUIDE	DSNTIPA0	89
APPL REGISTRATION TABLE	DSNTIPZ	216
APPLICATION DBRM	DSNTIPT	107
APPLICATION ENCODING	DSNTIPF	160
APPLICATION LOAD	DSNTIPT	107
ARCHIVE LOG FREQ	DSNTIPL	192
ARCHIVE LOG RACF	DSNTIPP	184
ART/ORT ESCAPE CHARACTER	DSNTIPZ	216
ASCII CODED CHAR SET	DSNTIPF	160
ASSEMBLER	DSNTIPG	123
ASSISTANT	DSNTIPK	101
AUDIT TRACE	DSNTIPN	149
AUTH AT HOP SITE	DSNTIP5	210
AUTH MEMBER	DSNTIPM	188
AUTH SEQUENCE	DSNTIPM	188
AUTO BIND	DSNTIPO	153
AUTO START	DSNTIPI	174
BACKOUT DURATION	DSNTIPL	192
BIND NEW PACKAGE	DSNTIPP	184
BLOCK SIZE	DSNTIPA	197
BP0 - BP25	DSNTIP1	143
BP26 - BP49	DSNTIP2	145
BP8K0-BP8K9	DSNTIP6	147
BP16K0-BP16K9	DSNTIP6	147
BP32K-BP32K9	DSNTIP6	147
BUFFER POOL SIZE	DSNTIPC	222
C COMPILER	DSNTIPU	112
C COMPILER MESSAGES	DSNTIPU	112
C LIBRARY HEADERS	DSNTIPU	112
C LINK EDIT STUBS	DSNTIPU	112
C DYNAMIC RUNTIME	DSNTIPU	112
C PRE-LINK MESSAGES	DSNTIPU	112
C PROGRAM NAME	DSNTIPU	112
CACHE DYNAMIC SQL	DSNTIP4	168
CATALOG ALIAS	DSNTIPA2	96
CATALOG DATA	DSNTIPA	197
CHECKPOINT FREQ	DSNTIPL	192
CICS COBOL LIBRARY	DSNTIPW	127
CICS COBOL II LIBRARY	DSNTIPW	127
CICS LOAD LIBRARY	DSNTIPW	127
CICS MACRO LIBRARY	DSNTIPW	127
CICS PL/I LIBRARY	DSNTIPW	127
CLIST LIBRARY	DSNTIPT	107
COBOL TYPE	DSNTIPY	219

Table 30. Panel fields (continued)

Panel Field Name	Panel	Page
CODE STORAGE SIZE	DSNTIPC	222
COLUMNS	DSNTIPD	130
COMMAND PREFIX	DSNTIPM	188
COMMAND REFERENCE	DSNTIPA0	89
COMMAND SCOPE	DSNTIPM	188
COMPACT DATA	DSNTIPA	197
CONTRACT THREAD STG	DSNTIPE	138
CONTROL ALL APPLICATIONS	DSNTIPZ	216
COORDINATOR	DSNTIPK	101
COPY 1 NAME	DSNTIPH	103
COPY 2 NAME	DSNTIPH	103
COPY 1 PREFIX	DSNTIPH	103
COPY 2 PREFIX	DSNTIPH	103
CPP AUTO CALL LIBRARY	DSNTIPU	112
CPP CLASS LIBRARY	DSNTIPU	112
CPP CLASS LIB HEADERS	DSNTIPU	112
CPP COMPILER	DSNTIPU	112
CPP COMPILER MESSAGES	DSNTIPU	112
CPP DYNAMIC RUNTIME	DSNTIPU	112
CPP LINK EDIT STUBS	DSNTIPU	112
CPP PRE-LINK MESSAGES	DSNTIPU	112
CPP PROCEDURE LIBRARY	DSNTIPG	123
CPP PROGRAM NAME	DSNTIPU	112
CPP STANDARD HEADERS	DSNTIPU	112
CROSS MEMORY	DSNTIPJ	179
CURRENT DEGREE	DSNTIP4	168
DATASET STATS TIME	DSNTIPN	149
DATA SET NAME(MEMBER)	DSNTIPA1	90
DATA SET STORAGE SIZE	DSNTIPC	222
DATA SHARING	DSNTIPA1	90
DATABASE PROTOCOL	DSNTIP5	210
DATABASES	DSNTIPE	138
DATABASES	DSNTIPD	130
DATE FORMAT	DSNTIP4	168
DBADM CREATE AUTH	DSNTIPP	184
DBRM LIBRARY	DSNTIPT	107
DB2 GENERIC LUNAME	DSNTIPR	205
DB2 LOCATION NAME	DSNTIPR	205
DB2 NETWORK LUNAME	DSNTIPR	205
DB2 NETWORK PASSWORD	DSNTIPR	205
DB2 PROC NAME	DSNTIPX	214
DDF STARTUP OPTION	DSNTIPR	205
DDF THREADS	DSNTIPR	205
DEADLOCK CYCLE	DSNTIPJ	179
DEADLOCK TIME	DSNTIPJ	179
DEALLOC PERIOD	DSNTIPA	197
DECIMAL ARITHMETIC	DSNTIPF	160
DECIMAL POINT IS	DSNTIPF	160
DECLARATION LIBRARY	DSNTIPT	107
DEF ENCODING SCHEME	DSNTIPF	160
DEFAULT BUFFER POOL FOR USER DATA	DSNTIP1	143
DEFAULT BUFFER POOL FOR USER	DSNTIP1	143
INDEXES		
DEFINE CATALOG	DSNTIPA2	96

Table 30. Panel fields (continued)

	Panel Field Name	Panel	Page
	DESCRIBE FOR STATIC	DSNTIPF	160
	DEVICE TYPE 1	DSNTIPA	197
	DEVICE TYPE 2	DSNTIPA	197
	DIST SQL STR DELIMTR	DSNTIPF	160
	DISCONNECT IRLM	DSNTIPJ	179
	DL/I BATCH TIMEOUT	DSNTIPI	174
	DPROP SUPPORT	DSNTIPO	153
	DRDA® PORT	DSNTIP5	210
	DSMAX	DSNTIPC	222
	EBCDIC CODED CHAR SET	DSNTIPF	160
I	EDMPPOOL DATA SPACE MAX	DSNTIPC	222
	EDMPPOOL DATA SPACE SIZE	DSNTIPC	222
	EDMPPOOL STORAGE SIZE	DSNTIPC	222
I	EVALUATE UNCOMMITTED	DSNTIP4	168
	EXECUTED STMTS	DSNTIPD	130
	EXIT LIBRARY	DSNTIPT	107
	EXPLAIN PROCESSING	DSNTIPO	153
	EXTENDED SECURITY	DSNTIPR	205
	EXTRA BLOCKS REQ	DSNTIP5	210
	EXTRA BLOCKS SRV	DSNTIP5	210
	Fortran COMPILER	DSNTIPG	123
	Fortran LINK EDIT	DSNTIPG	123
I	FREQUENCY TYPE	DSNTIPL	192
I	FROM RELEASE	DSNTIPA1	90
	GDDM LOAD MODULES	DSNTIPW	127
	GDDM MACLIB	DSNTIPW	127
	GROUP ATTACH	DSNTIPK	101
	GROUP NAME	DSNTIPK	101
	IBM COBOL COMPILER	DSNTIPQ	120
	IBM COBOL LINK EDIT	DSNTIPQ	120
	IBM COBOL PRELINK MSGS	DSNTIPQ	120
	IBM COBOL RUNTIME	DSNTIPQ	120
	IDLE THREAD TIMEOUT	DSNTIPR	205
I	IMMEDIATE WRITE	DSNTIP4	168
	IMS BMP TIMEOUT	DSNTIPI	174
	IMS RESLIB	DSNTIPW	127
	INCLUDE LIBRARY	DSNTIPT	107
	INPUT MEMBER NAME	DSNTIPA1	90
	INSTALL DD CONTROL SUPT.	DSNTIPZ	216
	INSTALL IRLM	DSNTIPI	174
	INSTALL TYPE	DSNTIPA1	90
	INSTALLATION GUIDE	DSNTIPA0	89
	IRLM LOAD LIBRARY	DSNTIPT	107
	IRLM LOCK MAXIMUM SPACE	DSNTIPC	222
	IRLM XCF GROUP NAME	DSNTIPJ	179
	ISPF ISPLINK MODULE	DSNTIPW	127
	IVP DATA LIBRARY	DSNTIPT	107
	LANGUAGE DEFAULT	DSNTIPF	160
I	LARGE EDM BETTER FIT	DSNTIP4	168
I	LEVELID UPDATE FREQ	DSNTIPL	192
	LE/370 LINK EDIT LIB	DSNTIPG	123
I	LE/370 RUNTIME LIBRARY	DSNTIPG	123
I	LIMIT BACKOUT	DSNTIPL	192
	LINK LIST ENTRY	DSNTIPM	188

Table 30. Panel fields (continued)

Panel Field Name	Panel	Page
LINK LIST LIBRARY	DSNTIPT	107
LINK LIST SEQUENCE	DSNTIPM	188
LOAD DISTRIBUTION	DSNTIPT	107
LOAD LIBRARY	DSNTIPT	107
LOCAL DATE LENGTH	DSNTIP4	168
LOCALE LC_CTYPE	DSNTIPF	160
LOCAL TIME LENGTH	DSNTIP4	168
LOCK ENTRY SIZE	DSNTIPJ	179
LOCKS PER TABLE(SPACE)	DSNTIPJ	179
LOCKS PER USER	DSNTIPJ	179
LOG APPLY STORAGE	DSNTIPL	192
MACRO LIBRARY	DSNTIPT	107
MANAGE THREAD STORAGE	DSNTIPE	138
MAX ABEND COUNT	DSNTIPX	214
MAX BATCH CONNECT	DSNTIPE	138
MAX DEGREE	DSNTIP4	168
MAX KEPT DYN STMTS	DSNTIPE	138
MAX REMOTE ACTIVE	DSNTIPE	138
MAX REMOTE CONNECTED	DSNTIPE	138
MAX TSO CONNECT	DSNTIPE	138
MAX TYPE 1 INACTIVE	DSNTIPR	205
MAX USERS	DSNTIPE	138
MAXIMUM ECSA	DSNTIPJ	179
MAXIMUM LE TOKENS	DSNTIP7	136
MEMBER IDENTIFIER	DSNTIPJ	179
MEMBER NAME	DSNTIPK	101
MINIMUM DIVIDE SCALE	DSNTIPF	160
MIXED DATA	DSNTIPF	160
MONITOR SIZE	DSNTIPN	149
MONITOR TRACE	DSNTIPN	149
NUMBER OF COPIES	DSNTIPH	103
NUMBER OF LOGS	DSNTIPL	192
NUMBER OF TCBS	DSNTIPX	214
OBJT REGISTRATION TABLE	DSNTIPZ	216
OPTIMIZATION HINTS	DSNTIP4	168
OUTPUT BUFFER	DSNTIPL	192
OUTPUT MEMBER NAME	DSNTIPA1	90
PACKAGE AUTH CACHE	DSNTIPP	184
PACKAGE LISTS	DSNTIPD	130
PACKAGE STATEMENTS	DSNTIPD	130
PACKAGES	DSNTIPD	130
PARAMETER MODULE	DSNTIPO	153
PERMANENT UNIT NAME	DSNTIPA2	96
PLAN AUTH CACHE	DSNTIPP	184
PLAN STATEMENTS	DSNTIPD	130
PLANS	DSNTIPD	130
PL/I COMPILER	DSNTIPG	123
PL/I DYN RUNTIME BASE	DSNTIPG	123
PL/I DYN RUNTIME COMMON	DSNTIPG	123
PL/I LINK EDIT BASE	DSNTIPG	123
PL/I LINK EDIT COMMON	DSNTIPG	123
POOL THREAD TIMEOUT	DSNTIP5	210
PREFIX	DSNTIPA1	90
PRIMARY QUANTITY	DSNTIPA	197

Table 30. Panel fields (continued)

Panel Field Name	Panel	Page
PROC NAME	DSNTIPI	174
QUIESCE PERIOD	DSNTIPA	197
READ COPY2 ARCHIVE	DSNTIPO	153
READ TAPE UNITS	DSNTIPA	197
RECALL DATABASE	DSNTIPO	153
RECALL DELAY	DSNTIPO	153
RECOMMENDED REAL STORAGE	DSNTIPC	222
RECORDING MAX	DSNTIPA	197
REGISTRATION DATABASE	DSNTIPZ	216
REGISTRATION OWNER	DSNTIPZ	216
RELEASE LOCKS	DSNTIP4	168
REMOTE LOCATION	DSNTIPY	219
REQUIRE FULL NAMES	DSNTIPZ	216
RESOURCE AUTHID	DSNTIPP	184
RESOURCE TIMEOUT	DSNTIPI	174
RESYNC INTERVAL	DSNTIPR	205
RESYNC PORT	DSNTIP5	210
RETAINED LOCK TIMEOUT	DSNTIPI	174
RETENTION PERIOD	DSNTIPA	197
RID POOL SIZE	DSNTIPC	222
RLF AUTO START	DSNTIPO	153
RLST ACCESS ERROR	DSNTIPO	153
RLST ACCESS ERROR	DSNTIPR	205
RLST NAME SUFFIX	DSNTIPO	153
RO SWITCH CHKPTS	DSNTIPL	192
RO SWITCH TIME	DSNTIPL	192
ROUTINE AUTH CACHE	DSNTIPP	184
SAMPLE LIBRARY	DSNTIPT	107
SECONDARY QTY	DSNTIPA	197
SEQUENTIAL CACHE	DSNTIPE	138
SITE TYPE	DSNTIPO	153
SMF ACCOUNTING	DSNTIPN	149
SMF STATISTICS	DSNTIPN	149
SOM DLL IMPORT LIBRARY	DSNTIPQ	120
SORT LIBRARY	DSNTIPW	127
SORT POOL SIZE	DSNTIPC	222
SQL STRING DELIMITER	DSNTIPF	160
START IRLM CTRACE	DSNTIPI	174
STATISTICS HISTORY	DSNTIPO	153
STATISTICS ROLLUP	DSNTIPO	153
STATISTICS SYNC	DSNTIPN	149
STATISTICS TIME	DSNTIPN	149
STD SQL LANGUAGE	DSNTIP4	168
STRING DELIMITER	DSNTIPF	160
SUBSYSTEM MEMBER	DSNTIPM	188
SUBSYSTEM NAME	DSNTIPI	174
SUBSYSTEM NAME	DSNTIPM	188
SUBSYSTEM SEQUENCE	DSNTIPM	188
SUFFIX	DSNTIPA1	90
SUPPRESS SOFT ERRORS	DSNTIPM	188
SYSTEM ADMIN 1	DSNTIPP	184
SYSTEM ADMIN 2	DSNTIPP	184
SYSTEM LOB VALUE STORAGE	DSNTIP7	136
SYSTEM MACLIB	DSNTIPW	127



Table 30. Panel fields (continued)

Panel Field Name	Panel	Page
SYSTEM OPERATOR 1	DSNTIPP	184
SYSTEM OPERATOR 2	DSNTIPP	184
SYSTEM PROCEDURES	DSNTIPW	127
TABLES	DSNTIPD	130
TABLES IN STMT	DSNTIPD	130
TABLE SPACES	DSNTIPD	130
TCP/IP ALREADY VERIFIED	DSNTIP5	210
TCP/IP KEEPALIVE	DSNTIP5	210
TEMP CLIST LIBRARY	DSNTIPT	107
TEMP 4K DATA SETS	DSNTIPD	130
TEMP 4K SPACE	DSNTIPD	130
TEMP 32K DATA SETS	DSNTIPD	130
TEMP 32K SPACE	DSNTIPD	130
TEMPORARY UNIT NAME	DSNTIPA2	96
TIME FORMAT	DSNTIP4	168
TIME TO AUTOSTART	DSNTIPI	174
TIMEOUT VALUE	DSNTIPX	214
TIMESTAMP ARCHIVES	DSNTIPH	103
TOTAL MAIN STORAGE	DSNTIPC	222
TOTAL STORAGE BELOW 16MB	DSNTIPC	222
TRACE AUTO START	DSNTIPN	149
TRACE SIZE	DSNTIPN	149
TRACKER SITE	DSNTIPO	153
U LOCK FOR RR/RS	DSNTIPI	174
UNICODE CCSID	DSNTIPF	160
UNKNOWN AUTHID	DSNTIPP	184
UNREGISTERED DDL DEFAULT	DSNTIPZ	216
UPDATE PART KEY COLS	DSNTIP4	168
UPDATE RATE	DSNTIPL	192
UR CHECK FREQ	DSNTIPL	192
UR LOG WRITE CHECK	DSNTIPL	192
USE FOR DYNAMICRULES	DSNTIPF	160
USE PROTECTION	DSNTIPP	184
USER LOB VALUE STORAGE	DSNTIP7	136
UTILITY GUIDE AND REFERENCE	DSNTIPA0	89
UTILITY CACHE OPTION	DSNTIPE	138
UTILITY TIMEOUT	DSNTIPI	174
VARCHAR FROM INDEX	DSNTIP4	168
VIEWS	DSNTIPD	130
VOLUME SERIAL 1	DSNTIPA2	96
VOLUME SERIAL 2	DSNTIPA2	96
VOLUME SERIAL 3	DSNTIPA2	96
VOLUME SERIAL 4	DSNTIPA2	96
VOLUME SERIAL 5	DSNTIPA2	96
VOLUME SERIAL 6	DSNTIPA2	96
VOLUME SERIAL 7	DSNTIPA2	96
VS COBOL COMPILER	DSNTIPQ	120
VS COBOL LINK EDIT	DSNTIPQ	120
VS COBOL II COMPILER	DSNTIPQ	120
VS COBOL II LINK EDIT	DSNTIPQ	120
WLM ENVIRONMENT	DSNTIPX	214
WLM PROC NAME	DSNTIPX	214
WORK FILE DB	DSNTIPK	101
WORKING STORAGE SIZE	DSNTIPC	222

*Table 30. Panel fields (continued)*

<b>Panel Field Name</b>	<b>Panel</b>	<b>Page</b>
WRITE TO OPER	DSNTIPA	197
WTO ROUTE CODES	DSNTIPO	153
WTOR ROUTE CODE	DSNTIPA	197
X LOCK FOR SEARCHED U/D	DSNTIPI	174

# Online book data set names panel: DSNTIPA0

```
DSNTIPA0                                ONLINE BOOK DATA SET NAMES
===>  _

                                         Welcome to DB2 Version 7

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.DSNAP0G1.BOOK
2  COMMAND REFERENCE       ===> DSNHELP.DSNCR0G1.BOOK
3  INSTALLATION GUIDE      ===> DSNHELP.DSNIG0G1.BOOK
4  UTILITY GUIDE AND REF   ===> DSNHELP.DSNUG0G1.BOOK

PRESS:  ENTER to continue  RETURN to exit  HELP for more information
```

Figure 5. Online book data set names panel: DSNTIPA0

## 1. APPL PROG AND SQL GUIDE

Acceptable values: Valid data set name ending in .BOOK. See page 93.  
Default: DSNHELP.DSNAP0G1.BOOK  
DSNZPxxx: none

## 2. COMMAND REFERENCE

Acceptable values: Valid data set name ending in .BOOK. See page 93.  
Default: DSNHELP.DSNCR0G1.BOOK  
DSNZPxxx: none

## 3. INSTALLATION GUIDE

Acceptable values: Valid data set name ending in .BOOK. See page 93.  
Default: DSNHELP.DSNIG0G1.BOOK  
DSNZPxxx: none

## 4. UTILITY GUIDE AND REF

Acceptable values: Valid data set name ending in .BOOK. See page 93.  
Default: DSNHELP.DSNUG0G1.BOOK  
DSNZPxxx: 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 8 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 2 values for migration only: the release you are migrating from,
and a data set and member name. This is the name used from a previous Installation/
Migration from field 7 below:

 3  FROM RELEASE          ===>          V5 or V6
 4  DATA SET NAME(MEMBER)===>

Enter name of your input data sets (SDSNLOAD, SDSNMACS, SDSNSAMP, SDSNCLST):
 5  PREFIX                ===> DSN510
 6  SUFFIX                ===>

Enter to set or save panel values (by reading or writing the named members):

 7  INPUT MEMBER NAME     ===> DSNTIDXA  Default parameter values
 8  OUTPUT MEMBER NAME    ===>          Save new values entered on panels

PRESS:  ENTER to continue  RETURN to exit  HELP for more information
```

Figure 6. 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 Version 6 or Version 5 to DB2 for OS/390 and z/OS Version 7. You are required to fill in field 4 (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 Figure 11 on page 103, Figure 17 on page 131, Figure 18 on page 136, and Figure 29 on page 184 for more 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 7. 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 *DB2 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 8. 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. FROM RELEASE

Acceptable values:	V5, V6
Default:	none
DSNZPxxx:	none

Specify whether you are migrating from Version 5 or Version 6.

### 4. 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 Version 6 or Version 5. 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(DSNTID51)`

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'CO'. 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.

## 5. PREFIX

Acceptable values:	1-18 characters
Default:	DSN710
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 7 (INPUT MEMBER NAME), the prefix for fields 1, 2, 3, and 15, and the prefix and suffix for fields 4 through 14 on installation panel DSNTIPT on page 107 are set. If all these data sets do not have the same prefix and suffix, you can change them on installation panel DSNTIPT.

## 6. 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	<i>prefix</i> .SDSNIVPD	

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 *DSN710* in field 4, and leave field 5 blank.

## 7. 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 4). 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 7 then the INPUT MEMBER NAME is the OUTPUT MEMBER NAME used when migrating the first member of the data sharing group to Version 7. The DATA SET NAME(MEMBER) field must specify a member containing the DB2 Version 6 or Version 5 values at your site. This member is applied last and overrides the CLIST values established by the member specified in field 7.

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 8) last time.

The following data set names are generated with the *prefix* and *suffix* values from fields 5 and 6 only when the input member name for field 7 is DSNTIDXA.

Table 31. Resulting data set names when using prefix and suffix parameters

Default Library Name	CLIST Edited Library Name
<i>prefix.DBRMLIB.DATA</i>	<i>prefix.DBRMLIB.DATA.suffix</i>
<i>prefix.RUNLIB.LOAD</i>	<i>prefix.RUNLIB.LOAD.suffix</i>
<i>prefix.SRCLIB.DATA</i>	<i>prefix.SRCLIB.DATA.suffix</i>
<i>prefix.SDSNDBRM</i>	<i>prefix.SDSNDBRM.suffix</i>
<i>prefix.SDSNLINK</i>	<i>prefix.SDSNLINK.suffix</i>
<i>prefix.SDSNLOAD</i>	<i>prefix.SDSNLOAD.suffix</i>
<i>prefix.SDSNMACS</i>	<i>prefix.SDSNMACS.suffix</i>
<i>prefix.ADSNLOAD</i>	<i>prefix.ADSNLOAD.suffix</i>
<i>prefix.SDSNMACS</i>	<i>prefix.SDSNMACS.suffix</i>
<i>prefix.SDSNSAMP</i>	<i>prefix.SDSNSAMP.suffix</i>
<i>prefix.SDSNCLST</i>	<i>prefix.SDSNCLST.suffix</i>
<i>prefix.SDSNIVPD</i>	<i>prefix.SDSNIVPD.suffix</i>
<i>prefix.SDSNC.H</i>	<i>prefix.SDSNC.H</i>
<i>prefix.SDXRRESL</i>	<i>prefix.SDXRRESL.suffix</i>

## 8. 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 and you will not have the values available for future updates. 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 5 and 6, 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 8) 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 107 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 231.

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 9. Data parameters panel: DSNTIPA2

### 1. CATALOG ALIAS

Acceptable values:	1-8 characters
Default:	DSNCAT
Update:	see Part 2 (Volume 1) of <i>DB2 Administration Guide</i>
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. **Recommendation:** 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 Part 2 (Volume 1) of *DB2 Administration Guide*. It is best to use the same integrated catalog facility catalog

alias for DB2 for OS/390 and z/OS Version 7 that you used for Version 6 because Version 7 uses many of your Version 6 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 32 for a list of available options.

Table 32. DB2 data sets integrated catalog facility catalog options

DB2 Data Sets	Options
DB2 directory (DSNDB01) (VSAM) DB2 catalog (DSNDB06) (VSAM) Default database (DSNDB04) (VSAM) Work file database (VSAM)	You must catalog these data sets in the primary integrated catalog facility catalog.
Active logs (VSAM) 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 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.bbbbbbbb.xxxxxxxx.y0001.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.
- bbbbbbb 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 Part 2 (Volume 1) of <i>DB2 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 edits job DSNTIJCA which, when run, 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 231
DSNZPxxx:	none

Specify the volume serial numbers for the data sets defined by installation or migration. 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 227 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)**

**(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 *prefix.NEW.SDSNTEMP* and *prefix.NEW.SDSNSAMP*.

**(VOLUME SERIAL 2)**

**(Field 4 on Panel DSNTIPA2)** Non-VSAM data. It is used for *prefix.DBRMLIB.DATA*, *prefix.RUNLIB.LOAD*, and *prefix.SRCLIB.DATA*. The default is DSNV01.

**(VOLUME SERIAL 3)**

**(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)**

**(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)**

**(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)**

**(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)**

**(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 231
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 231
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 *DB2 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 10. 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.

**Recommendation:** 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 the recommendation is 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 DB0G. See *DB2 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 and z/OS Version 7 adopts your Version 6 or Version 5 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          NO, YES or EXT (Extended date format)

PRESS: ENTER to continue  RETURN to exit  HELP for more information
```

Figure 11. System resource data set names: DSNTIPH

## 1. COPY 1 NAME

Acceptable values:	valid data set name; 1-33 characters
Default:	DSNCAT.BSDS01 or DSNCAT.DSN1.BSDS01
Update:	see Update on page 231; not during 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.*Annnonnnn* or DSNCAT.DSN1.BSDSx.*Annnonnnn* 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.
- *Annnonnnn* is generated internally.

For the definition of a valid data set name, see page 93.

## 2. COPY 2 NAME

Acceptable values:	valid data set name; 1-33 characters
Default:	DSNCAT.BSDS02 or DSNCAT.DSN1.BSDS02
Update:	see Update on page 231; not during 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 93.

## 3. NUMBER OF COPIES

Acceptable values:	1-2
Default:	2
Update:	see Update on page 231; not during 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 DSNCAT.DSN1.LOGCOPY1
Update:	see Update on page 231; not during 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.*Annnonnnn* or DSNCAT.DSN1.LOGCOPYx.*Annnonnnn* 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 93.

## 5. COPY 2 PREFIX

Acceptable values:	valid data set name prefix; 1-30 characters
Default:	DSNCAT.LOGCOPY2 or DSNCAT.DSN1.LOGCOPY2
Update:	see Update on page 231; not during 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 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 93.

## 8. COPY 2 PREFIX

Acceptable values:	valid data set name prefix; 1-35 characters
Default:	DSNCAT.ARCHLOG2 or 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:	NO, YES, EXT
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 NO, the archive data set name should not contain a timestamp.

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.Thhmsst,*

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

If you specify EXT, the archive data set name should contain a timestamp with an **extended** date component in the format:

*.Dyyyyddd.*

A value of EXT in this field causes the lengths of the values entered for field COPY 1 PREFIX and field COPY 2 PREFIX to be audited to ensure that neither exceeds 17 bytes (19 bytes for other settings of TIMESTAMP ARCHIVES).

---

## 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 33 summarizes these selections.

*Table 33. 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
14 IVP DATA LIBRARY  ====> prefix.SDSNIVPD.suffix
15 INCLUDE LIBRARY    ====> prefix.SDSNC.H
PRESS: ENTER to continue RETURN to exit HELP for more information

```

Figure 12. 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 90), 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 34. 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 93
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 93
Default:	<i>prefix.NEW.SDSNSAMP</i>
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 93
Default:	<i>prefix.NEW.SDSNCLST</i>
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 235 for more information. These fields must not be blank.

### 4. APPLICATION DBRM

Acceptable values:	valid data set name: see page 93
Default:	<i>prefix.DBRMLIB.DATA.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 93
Default:	<i>prefix.RUNLIB.LOAD.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 93
Default:	<i>prefix.SRCLIB.DATA.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 93
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 93
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 93
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 93
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 93
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 93
Default:	<i>prefix.SDSNDBRM.suffix</i>
Update:	cannot change during update
DSNZPxxx:	none

Specify the name of the library where the DBRMs shipped with DB2 are placed.

## 13. IRLM LOAD LIBRARY

Acceptable values:	valid data set name: see page 93
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.

#### 14. IVP DATA LIBRARY

Acceptable values:	valid data set name: see page 93
Default:	<i>prefix.SDSNIVPD.suffix</i>
Update:	cannot change during update
DSNZPxxx:	none

The IVP DATA LIBRARY indicates the data set name of SDSNIVPD which is the SMP/E target library for the DB2 installation verification procedure (IVP) input data and the expected output from the sample applications.

#### 15. INCLUDE LIBRARY

Acceptable values:	valid data set name: see page 93
Default:	<i>prefix.SDSNC.H</i>
Update:	cannot change during update
DSNZPxxx:	none

Specify the name of the Include library data set. This library is used in the DB2 language PROCS for C and C++.

## Data set names panel 2: DSNTIPU

The entries on this panel and the following panels, DSNTIPQ, DSNTIPG, and DSNTIPW, 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 35 as they appear in the jobs shipped with DB2. Change the names of the data sets if they are different at your site.

Table 35. Data set names used in jobs for products

Job	Data Set Name	Function
DSNTEJ1	SYS1.MACLIB	Assembler macro library
	SYS1.SORTLIB	DFSORT™ load modules (can be deleted if DFSORT is in link list)
DSNTEJ1L	CEE.SCEELKED	LE/370 linkage editor library
DSNTEJ1P	CEE.SCEERUN	PL/I dynamic runtime base library
	SYS1.SIBMLINK	PL/I dynamic runtime common library
DSNTEJ2A	SYS1.SORTLIB	DFSORT load modules (can be deleted if DFSORT is in link list)
DSNTEJ2C	CEE.SCEERUN	Language Environment dynamic runtime library
DSNTEJ2D	CEE.SCEERUN	Language Environment dynamic runtime library
DSNTEJ2E	CEE.SCEERUN	Language Environment dynamic runtime library
DSNTEJ2F	SYS1.MACLIB	Assembler macro library
	SYS1.VSF2FORT	VS Fortran runtime library
DSNTEJ2P	CEE.SCEERUN	PL/I dynamic runtime base library
	SYS1.SIBMLINK	PL/I dynamic runtime common library
DSNTEJ3C	CEE.SCEERUN	Language Environment dynamic runtime library
DSNTEJ3P	CEE.SCEERUN	PL/I dynamic runtime base library
	SYS1.SIBMLINK	PL/I dynamic runtime common library
DSNTEJ4C	IMSVS.RESLIB	IMS linkage editor library
DSNTEJ4P	CEE.SCEERUN	Language Environment dynamic runtime library
	IMSVS.RESLIB	IMS linkage editor library
	CEE.SCEELKED	PL/I linkage editor base library
	CEE.SCEERUN	PL/I dynamic runtime base library
	SYS1.SIBMBASE	PL/I linkage editor common library
	SYS1.SIBMLINK	PL/I dynamic runtime common library
DSNTEJ5A	CICS410.SDFHLOAD	CICS command translator and linkage editor
	CICS410.SDFHMAC	CICS macro library
	SYS1.MACLIB	Assembler macro library
DSNTEJ5C	CICS410.SDFHLOAD	CICS command translator and linkage editor library
	IGY.V1R2M0.SIGYCOMP(IGYCRCTL)	IBM COBOL for MVS & VM (V1R2) compiler load module See also the list of libraries used by DSNH CLIST in Chapter 2 of <i>DB2 Command Reference</i>
DSNTEJ5P	CICS410.SDFHLOAD	CICS command translator and linkage editor library
	CICS410.SDFHPLI	CICS PL/I linkage editor library
	CEE.SCEELKED	PL/I linkage editor base library
	SYS1.SIBMBASE	PL/I linkage editor common library
DSNTEJ6P	CEE.SCEERUN	Language Environment dynamic runtime library
DSNTEJ6D	CEE.SCEERUN	Language Environment dynamic runtime library
DSNTEJ6S	CEE.SCEERUN	Language Environment dynamic runtime library
DSNTEJ6T	CEE.SCEERUN	Language Environment dynamic runtime library

Table 35. Data set names used in jobs for products (continued)

Job	Data Set Name	Function
I DSNTEJ6W	CEE.SCEERUN	Language Environment dynamic runtime library
I DSNTEJ6Z	CEE.SCEERUN	Language Environment dynamic runtime library
DSNTEJ61	CEE.SCEERUN	Language Environment dynamic runtime library
DSNTEJ62	CEE.SCEERUN	Language Environment dynamic runtime library
DSNTEJ63	CEE.SCEERUN	Language Environment dynamic runtime library
DSNTEJ64	CEE.SCEERUN	Language Environment dynamic runtime library
DSNTEJ65	CEE.SCEERUN	Language Environment dynamic runtime library
DSNTEJ75	GDDM.SADMSAM	GDDM macro library
	GDDM.SADMMOD	GDDM load module library
DSNTIJMV <sup>1</sup>	SYS1.MACLIB	Assembler macro library
	CBC.SCBCCMP	C compiler library
	EDCDC120	C compiler load module
	EDC.SEDCDMSG(EDCMSGE)	C compiler message file
	CEE.SCEERUN(EDCPRLK)	C compiler prelink load module
	CEE.SCEELKED	Language Environment linkage editor library
	CEE.SCEEMSGP(EDCPMSG)	C prelink message file
	CEE.SCEECPP	C++ autolink library
	CBC.V3R1M1.SCBC3CPP	C++ compiler messages
	CBC.SCLBCPP	C++ class library
	CEE.SCEERUN	Language Environment dynamic runtime library
	CBC.SCLBH.HPP	C++ library headers
	CEE.SCEEH.H	C++ C library headers
	CEE.SCEEMSGP	Language Environment prelink editor messages
	CICS410.SDFHCOB	CICS library for OS/VS COBOL
	SYS1.COB2CICS	CICS COBOL II library
	CICS410.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.COB2COMP	COBOL II compiler load module
	SYS1.COB2LIB	COBOL II linkage editor library
	IGY.V1R2M0.SIGYCOMP(IGYCRCTL)	IBM COBOL for MVS & VM (V1R2) compiler load module
	CEE.SCEELKED	Language Environment linkage editor library
	CEE.SCEEMSGP	Language Environment prelink editor messages
	CEE.SCEERUN	Language Environment runtime library
	SOMMVS.V1R1M0.SGOSPLKD	SOMObjects® for MVS library
I	SOM.SGOSIMP	SOMObjects for DLL import library
	IMSVS.RESLIB	IMS linkage editor library
	ISP.SISPLOAD	ISPF ISPLINK module
	IEL0AA	PL/I compiler load module
	CEE.SCEERUN	PL/I dynamic runtime base library
	SYS1.SIBMLINK	PL/I dynamic runtime common library
	CEE.SCEELKED	PL/I linkage editor base library
	SYS1.SIBMBASE	PL/I linkage editor common library
	IEL.SIELCOMP(IEL1AA)	PL/I for MVS & VM

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.

Use this panel to define the data set names of your C or C/C++ program product libraries. The fields can be used to define AD/Cycle® C/370™ V3R2 (C only) or C/C++ for MVS/ESA™ V3R1, C/C++ for MVS/ESA V3R2, or C/C++ for OS/390. Even

though C and C++ share libraries in some versions of the products, DB2 requires that you define the data set names separately, depending on your need for C or C++. For more information about these libraries, consult the C or C++ program product documentation.

Data sets specified on this panel are used by the DB2 installation process to tailor three C/C++ language procedures that are generated by installation job DSNTIJMV:

- DSNHC can be used to prepare DB2 programs using C
- DSNHCPP can be used to prepare DB2 programs using C++
- DSNHCPP2 can be used to prepare a class and a client for a DB2 object-oriented program using C++
- DSNHCPPS contains the header file search path to be used by DSNHCPP and DSNHCPP2.

Data sets specified on this panel are used by the DB2 installation process to tailor the DB2 Interactive (DB2I) program preparation CLIST, DSNH.

If C is not installed on your system, accept the default values for fields 1-7 and do not run IVP jobs DSNTIJ2D, DSNTIJ2U, DSNTIJ6D, DSNTIJ6T, DSNTIJ63, DSNTIJ6W, DSNTIJ6Z, DSNTIJ64, DSNTIJ65, DSNTIJ71, DSNTIJ73, and DSNTIJ75. You should remove the DSNHC language proc from job DSNTIJMV. If you are migrating from DB2 for OS/390 and z/OS Version 5, remove the statement that renames DSNHC from the DSNTIJFV job. If C++ is not installed on your system, accept the default values for fields 8-17 and skip steps PH02US08 and PH02US09 of IVP jobs DSNTIJ2E or DSNTIJ2U. You should also remove all statements that refer to DAYNAME and MONTHNAME from part DSNTESU in the *prefix.SDSNSAMP* library if C++ is not available.

```

DSNTIPU          INSTALL DB2 - DATA SET NAMES PANEL 2
===>

Enter data set names below:
 1 C COMPILER          ===> CBC.SCBCCMP
 2 C COMPILER MESSAGES ===>
 3 C PRE-LINK MESSAGES ===> CEE.SCEEMSGP
 4 C LIBRARY HEADERS   ===> CEE.SCEEH.H
 5 C LINK EDIT STUBS   ===> CEE.SCEELKED
 6 C DYNAMIC RUNTIME   ===> CEE.SCEERUN
 7 C PROGRAM NAME      ===> CBCDRVR
 8 CPP COMPILER        ===> CBC.SCBCCMP
 9 CPP COMPILER MESSAGES ===>
10 CPP PRE-LINK MESSAGES ===> CEE.SCEEMSGP
11 CPP LINK EDIT STUBS  ===> CEE.SCEELKED
12 CPP DYNAMIC RUNTIME  ===> CEE.SCEERUN
13 CPP PROGRAM NAME     ===> CBCDRVR
14 CPP STANDARD HEADERS ===> CEE.SCEEH.H
15 CPP CLASS LIB HEADERS ===> CBC.SCLBH.HPP
16 CPP AUTO CALL LIBRARY ===> CEE.SCEECPP
17 CPP CLASS LIBRARY    ===> CBC.SCLBCPP

PRESS: ENTER to continue  RETURN to exit  HELP for more information

```

Figure 13. Data set names panel 2: DSNTIPU

## 1. C COMPILER

Acceptable values:	blank, or valid data set name: see page 93
Default:	CBC.SCBCCMP
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name of the C compiler library.

- For AD/Cycle C/370 V3R2, the SMP/E target data set name typically includes the qualifier SEDCDCMP.
- For C/C++ for MVS/ESA, the SMP/E target data set name typically includes the qualifier SCBC3CMP.
- For C/C++ for OS/390, the SMP/E target data set name typically includes the qualifier SCBCCMP.

If you specify a name in this field, a STEPLIB is added to each job provided by DB2 that uses this compiler. This field can be left blank if the compiler library is in the link list.

## 2. C COMPILER MESSAGES

Acceptable values:	blank, or valid data set name: see page 93
Default:	none
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name of the C compiler message file.

- For AD/CYCLE C/370 V3R2 and for C/C++ for MVS/ESA V3R1, the SMP/E target data set name typically includes the qualifier SEDCDMSG.
- For C/C++ for MVS/ESA V3R2 and for C/C++ for OS/390, the compiler messages reside elsewhere and this field should be left blank.

## 3. C PRE-LINK MESSAGES

Acceptable values:	blank, or valid data set name: see page 93
Default:	CEE.SCEEMSGP
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name for message issued by the IBM Language Environment<sup>®</sup> pre-linkage editor (EDCPRLK). The SMP/E target data set name typically includes the qualifier SCEEMSGP. 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 93
Default:	CEE.SCEEH.H
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name of the C language standard header files (for #INCLUDEs for facilities such as I/O or math functions).

- For AD/Cycle C/370 V3R2, the SMP/E target data set name typically includes the qualifier SEDCDHDR
- For C/C++ for MVS/ESA and for C/C++ for OS/390, these header files reside in the IBM Language Environment standard header file. The SMP/E target data set name typically includes the qualifier SCEEH.H.

## 5. C LINK EDIT STUBS

Acceptable values:	blank, or valid data set name: see page 93
Default:	CEE.SCEELKED
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name for objects required to link-edit C programs. The SMP/E target data set name typically includes the qualifier SCEELKED.

## 6. C DYNAMIC RUNTIME

Acceptable values:	blank, or valid data set name: see page 93
Default:	CEE.SCEERUN
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name of the C runtime library that contains the LE runtime modules. The SMP/E target data set name typically includes the qualifier SCEERUN.

## 7. C PROGRAM NAME

Acceptable values:	blank, or valid load module name: see page 93
Default:	CBCDRVR
Update:	cannot change during update
DSNZPxxx:	none

Specify the load module name for the compiler for your C program. Valid names are:

- For AD/Cycle C/370 V3R2, the compiler name is EDCDC120
- For C/C++ for MVS/ESA V3R1, the compiler name is CBC310
- For C/C++ for MVS/ESA V3R2, the compiler name is CBC320PP
- For C/C++ for OS/390, the compiler name alias is CBCDRVR

The name of the compiler will be used to configure your DSNHC language proc, the DSNH CLIST, and IVP jobs DSNTEJ2D, DSNTEJ2U, DSNTEJ6D, DSNTEJ6T, DSNTEJ71, DSNTEJ73, and DSNTEJ75.

## 8. CPP COMPILER

Acceptable values:	blank, or valid data set name: see page 93
Default:	CBC.SCBCCMP
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name of the C++ compiler library.

- For C/C++ for MVS/ESA, the data set name typically includes the qualifier SCBC3CMP.
- For C/C++ for OS/390, the data set name typically includes the qualifier SCBCCMP.

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++. If you specify a name in this field, a STEPLIB is added to the compile step of the DSNHCPP and DSNHCPP2 procs in job DSNTIJMV, and to the C++ portion of the DSNH CLIST. This field can be left blank if the compiler library is in the link list.

## 9. CPP COMPILER MESSAGES

Acceptable values:	blank, or valid data set name: see page 93
Default:	none
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name for C++ compiler messages file.

- For C/C++ for MVS/ESA V3R1, the data set name typically includes the qualifier SEDCDMSG.
- For C/C++ for MVS/ESA V3R2 and C++ for OS/390, the field should be left blank because the messages reside elsewhere.

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 DSNTIJ2E. 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 93
Default:	CEE.SCEEMSGP
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name for C++ pre-link messages file. The data set name typically includes the qualifier SCEEMSGP.

## 11. CPP LINK EDIT STUBS

Acceptable values:	blank, or valid data set name: see page 93
Default:	CEE.SCEELKED
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name of the C++ linkage editor library. The data set name typically includes the qualifier SCEELKED.

## 12. CPP DYNAMIC RUNTIME

Acceptable values:	blank, or valid data set name: see page 93
Default:	CEE.SCEERUN
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name for the C++ runtime library. The data set name typically includes the qualifier SCEERUN.

## 13. CPP PROGRAM NAME

Acceptable values:	blank, or valid data set name: see page 93
Default:	CBCDRVR
Update:	cannot change during update
DSNZPxxx:	none

Specify the C++ program name. The following are valid data set names:

- For C/C++ for MVS/ESA V3R1, the compiler name is CBC310PP
- For C/C++ for MVS/ESA V3R2, the compiler name is CBC320PP
- For C/C++ for OS/390, the compiler name alias is CBCDRVR

The compiler name will be used in the language procs, DSNHCPP and DSNHCPP2; the DSNHCPPS searchlist member; the DSNH CLIST; and IVP job DSNTEJ2E.

## 14. CPP STANDARD HEADERS

Acceptable values:	blank, or valid data set name: see page 93
Default:	CEE.SCEEH.H
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name of the C++ standard header files (for #INCLUDEs of facilities such as I/O and math functions). The data set name typically includes the qualifier SCEEH.H.

## 15. CPP CLASS LIB HEADERS

Acceptable values:	blank, or valid data set name: see page 93
Default:	CBC.SCLBH.H
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name of the C++ class library header files (for object-oriented programming).

- For C/C++ for MVS/ESA, the data set name typically includes the qualifier SCLB3H.H
- For C/C++ for OS/390, the data set name typically includes the qualifier SCLBH.H



## 16. CPP AUTO CALL LIBRARY

Acceptable values:	blank, or valid data set name: see page 93
Default:	CEE.SCEECPP
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name of the C++ auto call library or LE pre-linkage editor (EDCPRLK). This data set name typically includes the qualifier SCEECPP.

## 17. CPP CLASS LIBRARY

Acceptable values:	blank, or valid data set name: see page 93
Default:	CBC.SCLBCPP
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name of the C++ class library (for object-oriented programming) required during pre-linkedit processing.

- For C/C++ for MVS/ESA, the data set name typically includes the qualifier SCLB3CPP
- For C/C++ for OS/390, the data set name typically includes the qualifier SCLBCPP

---

## 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

3 VS COBOL II COMPILER

4 VS COBOL II LINK EDIT

5 IBM COBOL COMPILER

6 IBM COBOL RUNTIME

7 IBM COBOL PRE-LINK MSGS

8 IBM COBOL LINK EDIT

9 SOM DLL IMPORT LIBRARY

====>

====> SYS1.COBLIB

====>

====> SYS1.V1R4.COB2LIB

====>

====> CEE.SCEERUN

====> CEE.SCEMSGP

====> CEE.SCEELKED

====> SOM.SGOSIMP

PRESS: ENTER to continue RETURN to exit HELP for more information

Figure 14. Data set names panel 3: DSNTIPQ

If COBOL is not installed on your system, do not run jobs DSNTJ2C, DSNTJ3C, DSNTJ4C, DSNTJ5C, or DSNTJ6. If IBM COBOL is not installed on your system, do not run jobs DSNTJ61 and DSNTJ62.

### 1. VS COBOL COMPILER

Acceptable values:	blank, or valid data set name: see page 93
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. The current default for DB2 is IBM COBOL for MVS and VM.

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 355 for the specific changes.

### 2. VS COBOL LINK EDIT

Acceptable values:	valid data set name: see page 93
Default:	SYS1.COBLIB
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name of the COBOL linkage library. The current default for DB2 is IBM COBOL for MVS and VM.

### 3. VS COBOL II COMPILER

Acceptable values:	blank, or valid data set name: see page 93
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 93
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 93
Default:	none
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name of the COBOL compiler load module library.

### 6. IBM COBOL RUNTIME

Acceptable values:	blank, or valid data set name: see page 93
Default:	CEE.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 93
Default:	CEE.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 93
Default:	CEE.SCEELKED
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name of the COBOL link edit library.

## 9. SOM DLL IMPORT LIBRARY

Acceptable values:	blank, or valid data set name: see page 93
Default:	SOM.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 use IBM COBOL for MVS & VM Version 1 Release 2 or IBM COBOL Verion 2 Release 1 for preparing non-object-oriented application programs, blank out the data set name of the SOM DLL import library.

## 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 112.

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. See the table below to determine which fields need values.

Table 36. How DB2 recognizes your PL/I compiler

PL/I compiler	Fields with values	Fields left as defaults
PL/I MVS and VM	PL/I LINK EDIT BASE PL/I DYN RUNTIME BASE	PL/I LINK EDIT COMMON PL/I DYN RUNTIME COMMON
OS PL/I Version 1	PL/I LINK EDIT BASE	PL/I LINK EDIT COMMON PL/I DYN RUNTIME BASE PL/I DYN RUNTIME COMMON
OS PL/I Version 2	all	none

DSNTIPG

INSTALL DB2 - DATA SET NAMES PANEL 4

===>

Enter data set names below:

1

ASSEMBLER

====>

2

FORTTRAN COMPILER

====>

3

FORTTRAN LINK EDIT

====> SYS1.VSF2FORT

4

PL/I COMPILER

====>

5

PL/I LINK EDIT BASE

====> CEE.SCEELKED

6

PL/I LINK EDIT COMMON

====>

7

PL/I DYN RUNTIME BASE

====> CEE.SCEERUN

8

PL/I DYN RUNTIME COMMON

====>

9

CPP PROCEDURE LIBRARY

====> CBC.SCBCUTL

10

LE/370 RUNTIME LIBRARY

====> CEE.SCEERUN

11

LE/370 LINK EDIT LIB

====> CEE.SCEELKED

PRESS:

ENTER to continue

RETURN to exit

HELP for more information

Figure 15. Data set names panel 4: DSNTIPG

### 1. ASSEMBLER

Acceptable values:	blank, or valid data set name: see page 93
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 93
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 DSNTSJ2F. 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 93
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 93
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.

# 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. For  
# more information, see Table 36 on page 123.

If PL/I is not installed on your system, do not run jobs DSNTSJ1P, DSNTSJ2P, DSNTSJ3P, DSNTSJ4P, DSNTSJ5P, DSNTSJ6P, DSNTSJ6S, or DSNTSJ6U. You can use SPUFI or QMF to provide the listings of the sample tables and dynamic SQL examples that are provided in jobs DSNTSJ1P and DSNTSJ3P. 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 93
Default:	CEE.SCEELKED
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 93
Default:	none
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 93
Default:	CEE.SCEERUN
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 93
Default:	none
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 93
Default:	CBC.SCBCUTL
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 93
Default:	CEE.SCEERUN
Update:	cannot change during update
DSNZPxxx:	none

This field is used to tailor the JCL procedure that starts the stored procedures address space. 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).

## 11. LE/370 LINK EDIT LIB

Acceptable values:	valid data set name: see page 93
Default:	CEE.SCEELKED
Update:	cannot change during update
DSNZPxxx:	none

This field is used to tailor the sample job DSNTEJ1L which is used to create a DSNTEP2 load module.



# 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 35 on page 112 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.SISPLMOD

6

GDDM MACLIB

====>

GDDM.SADMSAM

7

GDDM LOAD MODULES

====>

GDDM.SADMMOD

8

CICS LOAD LIBRARY

====>

CICS410.SDFHLOAD

9

CICS MACRO LIBRARY

====>

CICS410.SDFHMAC

10

CICS COBOL LIBRARY

====>

CICS410.SDFHCOB

11

CICS COBOL II LIBRARY

====>

SYS1.COB2CICS

12

CICS PL/I LIBRARY

====>

CICS410.SDFHPL1

13

CICS EXCI LIBRARY

====>

CICS410.SDFHEXCI

PRESS:

ENTER to continue

RETURN to exit

HELP for more information

Figure 16. Data set names panel 5: DSNTIPW

## 1. SYSTEM MACLIB

Acceptable values:

valid data set name: see page 93

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 93

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 93
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 93
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 DSNTJEJ4C and DSNTJEJ4P.

### 5. ISPF ISPLINK MODULE

Acceptable values:	valid data set name: see page 93
Default:	ISP.SISPLOAD
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name of the ISPF load module library.

### 6. GDDM MACLIB

Acceptable values:	valid data set name: see page 93
Default:	GDDM.SADMSAM
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name of the GDDM macro library. This field and GDDM LOAD MODULES must both have a valid data set name or both be blank. The data set name specified in this field is included in the compile step SYSLIB DD concatenations of DSNHASM, DSNHC, DSNHCOB, DSNHCOB2, DSNHICOB, DSNHICB2, and DSNHPLI. The installation CLIST will only generate sample job DSNTJEJ75 if you specify a GDDM MACLIB name.

### 7. GDDM LOAD MODULES

Acceptable values:	valid data set name: see page 93
Default:	GDDM.SADMMOD
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name of the GDDM load module library. This field and GDDM MACLIB must both have a valid data set name or both be blank. The data set name specified in this field is included in the link-edit SYSLIB concatenations of DSNHASM, DSNHC, DSNHCOB, DSNHCOB2, DSNHICOB, DSNHICB2, and DSNHPLI.

## 8. CICS LOAD LIBRARY

Acceptable values:	valid data set name: see page 93
Default:	CICS410.SDFHLOAD
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name for the CICS load module library. If CICS is not used, blank out the CICS LOAD LIBRARY field.

## 9. CICS MACRO LIBRARY

Acceptable values:	valid data set name: see page 93
Default:	CICS410.SDFHMAC
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name for the CICS macro library.

## 10. CICS COBOL LIBRARY

Acceptable values:	valid data set name: see page 93
Default:	CICS410.SDFHCOB
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name for the CICS library for use by the COBOL programs.

## 11. CICS COBOL II LIBRARY

Acceptable values:	valid data set name: see page 93
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™.

## 12. CICS PL/I LIBRARY

Acceptable values:	valid data set name: see page 93
Default:	CICS410.SDFHPLI
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name for the CICS library for use by the PL/I programs.

## 12. CICS EXCI LIBRARY

Acceptable values:	valid data set name: see page 93
Default:	CICS410.SDFHEXCI
Update:	cannot change during update
DSNZPxxx:	none

Specify the data set name for the CICS library that contains the CICS EXCI load modules.

#  
#  
#  
#  
#  
#  
#

---

## Sizes panel 1: 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 231 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 most fields, 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 7 on page 26. One exception is the value for the work file database. To obtain this value, see “Disk requirements for the work file database” on page 29. If you have a large site, increase the values according to your needs.

**If you are migrating,** DB2 for OS/390 and z/OS Version 7 adopts your Version 6 or Version 5 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 231. You cannot actually change the values of these parameters.

```

DSNTIPD          INSTALL DB2 - SIZES PANEL 1
==> -

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    ==> 16     Size of 4K-page work space (megabytes)
14 TEMP 4K DATA SETS ==> 1    Number of data sets for 4K data
15 TEMP 32K SPACE   ==> 4      Size of 32K-page work space (megabytes)
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 17. Sizes panel 1: DSNTIPD

## 1. DATABASES

Acceptable values: 1-64000  
Default: 200  
Update: see Update on page 231; 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 231; 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 231; 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 231  
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 231; 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 231; 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 231; 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 231; 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 231; 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 231; 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 231; 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 231; 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:	1-2000000
Default:	16
Update:	see below; not during migration
DSNZPxxx:	none

Specify in megabytes the total size of the 4KB table spaces in the work file database.

Database DSNDB07 is used as temporary space for SQL statements and triggers 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 16 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 work file 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 work file table spaces to support large temporary files. For information about creating additional temporary work file table spaces, see Part 5 (Volume 2) of *DB2 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 255.

#### 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:	0-2000000
Default:	4
Update:	see field 11 or see below; not during migration
DSNZPxxx:	none

Specify in megabytes 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 3 (DSNTIP6).

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 3 (DSNTIP6).
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.

---

## Sizes panel 2: DSNTIP7

The entries on this panel establish limits for the amount of storage that can be used for storing large object (LOB) values.

```
DSNTIP7          INSTALL DB2 - SIZES PANEL 2
===> _

Check numbers and reenter to change:

  1  USER LOB VALUE STORAGE    ===> 2048    Max storage per user for LOB
                                           values in kilobytes
  2  SYSTEM LOB VALUE STORAGE  ===> 2048    Max storage per system for LOB
                                           values in megabytes
  3  MAXIMUM LE TOKENS         ===> 20      Maximum tokens at any time. 0-50

PRESS:  ENTER to continue  RETURN to exit  HELP for more information
```

Figure 18. Sizes panel 2: DSNTIP7

### 1. USER LOB VALUE STORAGE

Acceptable values: 1-2097152  
Default: 2048  
Update: option 10 on panel DSNTIPB  
DSNZPxxx: DSN6SYSP LOBVALA

The value in this field establishes an upper limit for the amount of storage each user can have for storing LOB values. The value specified gives the numbers of kilobytes.

### 2. SYSTEM LOB VALUE STORAGE

Acceptable values: 1-51200  
Default: 2048  
Update: option 10 on panel DSNTIPB  
DSNZPxxx: DSN6SYSP LOBVALS

The value in this field establishes an upper limit for the amount of storage per system that can be used for storing LOB values. The value specified gives the numbers of megabytes.

### 3. MAXIMUM LE TOKENS

Acceptable values: 0-50  
Default: 20  
Update: option 10 on panel DSNTIPB  
DSNZPxxx: DSN6SPRM LEMAX

This number represents the maximum number of LE tokens that are active at any time. If the value is zero, no tokens are available. A token is used each time one of the following functions is used:

- trigonometry functions (sin, sinh, asin, cos, cosh, acos, tan, tanh, atanh, atan, and atan2)

- degrees
- radians
- rand
- exp
- power
- log functions (log, and log10)
- upper
- lower
- translate
- ROUND\_TIMESTAMP
- TRUNC\_TIMESTAMP
- LAST\_DAY
- NEXT\_DAY
- ADD\_MONTHS

For details on these functions, see *DB2 SQL Reference*. DB2 may use a LE token to perform conversion from one CCSID to another CCSID.

---

## 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 remote DDF
                                     connections that are supported
 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 MAX KEPT DYN STMTS  ===> 5000    Maximum number of prepared dynamic
                                     statements saved past commit points
10 CONTRACT THREAD STG ===> NO      Periodically free unused thread stg
11 MANAGE THREAD STORAGE ===> NO    Manage thread stg to minimize size

PRESS:  ENTER to continue  RETURN to exit  HELP for more information
```

Figure 19. Thread management panel: DSNTIPE

### 1. DATABASES

Acceptable values: 1-800  
Default: 100  
Update: option 11 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 222 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 Part 5 (Volume 2) of *DB2 Administration Guide*.

### 2. MAX USERS

Acceptable values: 1-2000  
Default: 70  
Update: option 11 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.
- Each utility (each utility uses one thread, plus one thread for each subtask)

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.

#  
#  
#  
#  
#

Due to parallelism, DB2 utilities each use a minimum of one thread, plus an additional thread for each subtask. Therefore, a single utility might use many threads. Specify a thread value accordingly to accommodate parallelism within utilities. Consider using a value that is higher than the default or the value you specified in a previous version of DB2.

### 3. MAX REMOTE ACTIVE

Acceptable values:	0-1999
Default:	64
Update:	option 11 on panel DSNTIPB
DSNZPxxx:	DSN6SYSP MAXDBAT

Specify the maximum number of database access threads (DBATs) that can be active concurrently.

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. The action taken when a request for a new connection to DB2 is received and MAX REMOTE ACTIVE has been reached depends on whether ACTIVE or INACTIVE is specified for option DDF THREADS on panel DSNTIPR.

#### If DDF THREADS is...

##### Action taken is..

<b>ACTIVE</b>	If MAX REMOTE CONNECTED has not been reached, the allocation request is allowed but any further processing for the connection is queued waiting for an active database access thread to terminate.
<b>INACTIVE</b>	If MAX REMOTE CONNECTED has not been reached, the allocation request is allowed and is processed when DB2 can assign an unused database access thread slot to the connection.

**Setting max remote active to zero:** You can use a zero in this field 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.

#### 4. MAX REMOTE CONNECTED

Acceptable values:	0-150000
Default:	64
Update:	option 11 on panel DSNTIPB
DSNZPxxx:	DSN6SYSP CONDBAT

This field limits the total number of inbound DDF connections.

Specify the maximum number of concurrent remote connections. This value must be greater than or equal to MAX REMOTE ACTIVE. When a request to allocate a new connection to DB2 is received, and MAX REMOTE CONNECTED has been reached, the connection request is rejected. See Part 5 (Volume 2) of *DB2 Administration Guide* for more information.

#### 5. MAX TSO CONNECT

Acceptable values:	1-2000
Default:	40
Update:	option 11 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:	20
Update:	option 11 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 DB2 utility
- 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.

REBUILD INDEX processing uses DB2 connections and may cause message DSNU397I to be issued. If you receive message DSNU397I indicating the REBUILD INDEX utility is constrained, increase the number of concurrent connections.

# Due to parallelism, DB2 utilities each use a minimum of one thread, plus an  
# additional thread for each subtask. Therefore, a single utility might use many  
# threads. Specify a maximum concurrent connections value accordingly to  
# accomodate parallelism within utilities. Consider using a value that is higher than  
# the default or the value you specified in a previous version of DB2.

## 7. SEQUENTIAL CACHE

Acceptable values:	BYPASS, SEQ
Default:	BYPASS
Update:	option 11 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 bypasses the cache. If you specify SEQ, DB2 prefetch uses sequential access for read activity. Many sites gain a performance benefit by specifying SEQCACH. See Part 5 (Volume 2) of *DB2 Administration Guide* for a discussion of these considerations.

**Recommendation:** Specify SEQCACH if you have current disk devices with good cache sizes, especially if the units are Enterprise Storage System (ESS) or RAMAC® Virtual Array (RVA).

## 8. UTILITY CACHE OPTION

Acceptable values:	YES, NO
Default:	NO
Update:	option 11 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 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. MAX KEPT DYN STMTS

Acceptable values:	0-65535
Default:	5000
Update:	option 11 on panel DSNTIPB
DSNZPxxx:	DSN6SPRM MAXKEEPD

Specify the total number of prepared, dynamic SQL statements that can be saved 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 Part 6 of *DB2 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.

## 10. CONTRACT THREAD STG

Acceptable values:	YES, NO
Default:	NO
Update:	option 11 on panel DSNTIPB
DSNZPxxx:	DSN6SPRM CONTSTOR

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. If YES is specified for this parameter, DB2 will examine threads at commit points and periodically return storage to the operating system that is no longer in use. For best performance, this parameter should be NO. For subsystems that have many long-running threads and that are constrained on storage in the DBM1 address space, specifying YES can reduce the total storage used in the DBM1 address space.

## 11. MANAGE THREAD STORAGE

Acceptable values:	YES, NO
Default:	NO
Update:	option 13 on panel DSNTIPB
DSNZPxxx:	DSN6SPRM MINSTOR

Specify whether DB2 will use storage management algorithms that minimize the amount of working storage consumed by individual threads. For best performance, this parameter should be NO. For subsystems that have many long-running threads and are constrained on storage in the DBM1 address space, specifying YES can reduce the total storage used in the DBM1 address space.



## Buffer pool sizes panel 1: DSNTIP1

This is the first of three panels that lets you choose the size of your virtual buffer pools, the size of your hiperpools, and whether you want your buffer pools all in the primary address space (database services address space) or in an MVS data space. For information about the two other buffer pool sizes panels, see “Buffer pool sizes panel 2: DSNTIP2” on page 145 and “Buffer pool sizes panel 3: DSNTIP6” on page 147. For a complete description of tuning buffer pools and hiperpools, see Part 5 (Volume 2) of *DB2 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
====> _

1 DEFAULT BUFFER POOL FOR USER DATA   ==> BP0      BP0-BP49
2 DEFAULT BUFFER POOL FOR USER INDEXES ==> BP0      BP0-BP49

Enter sizes in number of pages, and D (Dataspace) or P (Primary) for VPTYPE.
  BUFFERPOOL  Hiperpool  VPTYPE      BUFFERPOOL  Hiperpool  VPTYPE
3 BP0  ==> 2000  ==> 0      ==> P    16 BP13 ==> 0      ==> 0      ==> P
4 BP1  ==> 0     ==> 0      ==> P    17 BP14 ==> 0      ==> 0      ==> P
5 BP2  ==> 0     ==> 0      ==> P    18 BP15 ==> 0      ==> 0      ==> P
6 BP3  ==> 0     ==> 0      ==> P    19 BP16 ==> 0      ==> 0      ==> P
7 BP4  ==> 0     ==> 0      ==> P    20 BP17 ==> 0      ==> 0      ==> P
8 BP5  ==> 0     ==> 0      ==> P    21 BP18 ==> 0      ==> 0      ==> P
9 BP6  ==> 0     ==> 0      ==> P    22 BP19 ==> 0      ==> 0      ==> P
10 BP7  ==> 0     ==> 0      ==> P    23 BP20 ==> 0      ==> 0      ==> P
11 BP8  ==> 0     ==> 0      ==> P    24 BP21 ==> 0      ==> 0      ==> P
12 BP9  ==> 0     ==> 0      ==> P    25 BP22 ==> 0      ==> 0      ==> P
13 BP10 ==> 0     ==> 0      ==> P    26 BP23 ==> 0      ==> 0      ==> P
14 BP11 ==> 0     ==> 0      ==> P    27 BP24 ==> 0      ==> 0      ==> P
15 BP12 ==> 0     ==> 0      ==> P    28 BP25 ==> 0      ==> 0      ==> P

PRESS:  ENTER to continue  RETURN to exit  HELP for more information
```

Figure 20. Buffer pool sizes panel 1: DSNTIP1

### 1.DEFAULT BUFFER POOL FOR USER DATA

Acceptable values: Any 4KB buffer pool names  
Default: BP0  
Update: see below  
DSNZPxxx: DSN6SYSP TBSBPOOL

Specify the default buffer pool to use for user table spaces. This must be a 4KB buffer pool.

### 2.DEFAULT BUFFER POOL FOR USER INDEXES

Acceptable values: Any 4KB buffer pool names  
Default: BP0  
Update: see below  
DSNZPxxx: DSN6SYSP IDXBPOOL

Specify the default buffer pool to use for indexes on user data. This must be a 4KB buffer pool.

### 3-28. BUFFERPOOL

Acceptable values:	If VPTYPE=P: for BP0, 56-400000; for BP1-BP25, 0-400000. If VPTYPE=D: for BP0, 56-8000000; for BP1-BP25, 0-8000000.
Default:	for BP0, 2000; for BP1-BP25, 0
Update:	ALTER BUFFERPOOL command
DSNZPxxx:	none

Specify the total number of 4KB buffers in the given virtual buffer pool (BP0-BP25).

### 3-28. 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-BP25). 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.

### 3-28. VPTYPE

Acceptable values:	P, D
Default:	P
Update:	ALTER BUFFERPOOL command
DSNZPxxx:	none

Specify the type of virtual buffer pool to be allocated.

P indicates that the virtual buffer pool is to be allocated in the DB2 database services address space.

D indicates that the virtual buffer pool is to be allocated in one or more data spaces associated with DB2.

The VPTYPE specification becomes effective the next time the virtual buffer pool is allocated. If you have already allocated the virtual buffer pool and later change the VPTYPE attribute, the new VPTYPE specification becomes “pending” until the next time the virtual buffer pool is allocated.

***The Effect of Allocating a Buffer Pool before Running DSNTIJTM:*** If you specify that you want a buffer pool to be allocated in a data space and if the buffer pool gets allocated before you run job DSNTIJTM, the data space attribute is made pending and you will have to reallocate the buffer pool to get the data space attribute to take effect. This is likely to be a consideration only for BP0, which is always allocated during DB2 restart.

## Buffer pool sizes panel 2: DSNTIP2

This is the second of the three panels that lets you choose the size of your virtual buffer pools, the size of your hiperpools, and whether you want your buffer pools all in the primary address space (database services address space) or in an MVS data space. The first panel is described in “Buffer pool sizes panel 1: DSNTIP1” on page 143, and the third panel is described in “Buffer pool sizes panel 3: DSNTIP6” on page 147. For a complete description of tuning buffer pools and hiperpools, see Part 5 (Volume 2) of *DB2 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), and D (Dataspace) or P (Primary) for VPTYPE.

      BUFFERPOOL  Hiperpool  VPTYPE
1 BP26 ==> 0      ==> 0      ==> P  13 BP38 ==> 0      ==> 0      ==> P
2 BP27 ==> 0      ==> 0      ==> P  14 BP39 ==> 0      ==> 0      ==> P
3 BP28 ==> 0      ==> 0      ==> P  15 BP40 ==> 0      ==> 0      ==> P
4 BP29 ==> 0      ==> 0      ==> P  16 BP41 ==> 0      ==> 0      ==> P
5 BP30 ==> 0      ==> 0      ==> P  17 BP42 ==> 0      ==> 0      ==> P
6 BP31 ==> 0      ==> 0      ==> P  18 BP43 ==> 0      ==> 0      ==> P
7 BP32 ==> 0      ==> 0      ==> P  19 BP44 ==> 0      ==> 0      ==> P
8 BP33 ==> 0      ==> 0      ==> P  20 BP45 ==> 0      ==> 0      ==> P
9 BP34 ==> 0      ==> 0      ==> P  21 BP46 ==> 0      ==> 0      ==> P
10 BP35 ==> 0      ==> 0      ==> P  22 BP47 ==> 0      ==> 0      ==> P
11 BP36 ==> 0      ==> 0      ==> P  23 BP48 ==> 0      ==> 0      ==> P
12 BP37 ==> 0      ==> 0      ==> P  24 BP49 ==> 0      ==> 0      ==> P

PRESS:  ENTER to continue  RETURN to exit  HELP for more information
```

Figure 21. Buffer pool sizes panel 2: DSNTIP2

### 1-24. BUFFERPOOL

Acceptable values: If VPTYPE=P: 0-400000. If VPTYPE=D: 0-8000000  
Default: 0  
Update: ALTER BUFFERPOOL command  
DSNZPxxx: none

Specify the total number or 4KB buffers in the given virtual buffer pool (BP26-BP49).

### 1-24. 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 (BP26-BP49). If you specify a size in this field, a hiperpool is 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.

#### 1-24. **VPTYPE**

Acceptable values:	P, D
Default:	P
Update:	ALTER BUFFERPOOL command
DSNZPxxx:	none

Specify the type of virtual buffer pool to be allocated.

'P' indicates that the virtual buffer pool is to be allocated in the DB2 database services address space.

'D' indicates that the virtual buffer pool is to be allocated in one or more data spaces associated with DB2.

The VPTYPE specification becomes effective the next time the virtual buffer pool is allocated. If you have already allocated the virtual buffer pool and later change the VPTYPE attribute, the new VPTYPE specification becomes "pending" until the next time the virtual buffer pool is allocated.

## Buffer pool sizes panel 3: DSNTIP6

This is the third of the three panels that let you choose the size of your virtual buffer pools, the size of your hiperpools, and whether you want your buffer pools all in the primary address space (database services address space) or in an MVS data space. The first panel is described in “Buffer pool sizes panel 1: DSNTIP1” on page 143 and the second panel is described in “Buffer pool sizes panel 2: DSNTIP2” on page 145. For a complete description of tuning buffer pools and hiperpools, see Part 5 (Volume 2) of *DB2 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.

```
DSNTIP6          INSTALL DB2 - BUFFER POOL SIZES - PANEL 3
====> _

Enter sizes in number of pages, and D (Dataspace) or P (Primary) for VPTYPE.

      BUFFERPOOL  Hiperpool  VPTYPE      BUFFERPOOL  Hiperpool  VPTYPE
1 BP8K0 ==> 0    ==> 0    ==> P  16 BP16K5 ==> 0    ==> 0    ==> P
2 BP8K1 ==> 0    ==> 0    ==> P  17 BP16K6 ==> 0    ==> 0    ==> P
3 BP8K2 ==> 0    ==> 0    ==> P  18 BP16K7 ==> 0    ==> 0    ==> P
4 BP8K3 ==> 0    ==> 0    ==> P  19 BP16K8 ==> 0    ==> 0    ==> P
5 BP8K4 ==> 0    ==> 0    ==> P  20 BP16K9 ==> 0    ==> 0    ==> P
6 BP8K5 ==> 0    ==> 0    ==> P  21 BP32K  ==> 24    ==> 0    ==> P
7 BP8K6 ==> 0    ==> 0    ==> P  22 BP32K1 ==> 0    ==> 0    ==> P
8 BP8K7 ==> 0    ==> 0    ==> P  23 BP32K2 ==> 0    ==> 0    ==> P
9 BP8K8 ==> 0    ==> 0    ==> P  24 BP32K3 ==> 0    ==> 0    ==> P
10 BP8K9 ==> 0    ==> 0    ==> P  25 BP32K4 ==> 0    ==> 0    ==> P
11 BP16K0 ==> 0    ==> 0    ==> P  26 BP32K5 ==> 0    ==> 0    ==> P
12 BP16K1 ==> 0    ==> 0    ==> P  27 BP32K6 ==> 0    ==> 0    ==> P
13 BP16K2 ==> 0    ==> 0    ==> P  28 BP32K7 ==> 0    ==> 0    ==> P
14 BP16K3 ==> 0    ==> 0    ==> P  29 BP32K8 ==> 0    ==> 0    ==> P
15 BP16K4 ==> 0    ==> 0    ==> P  30 BP32K9 ==> 0    ==> 0    ==> P

PRESS:  ENTER to continue  RETURN to exit  HELP for more information
```

Figure 22. Buffer pool sizes panel 3: DSNTIP6

### 1-30. BUFFERPOOL

Acceptable values:	If VPTYPE=P: for BP8K0-BP8K9, 0-200000;BP16K0-BP16K9, 0-100000; for BP32K-BP32K9, 0-50000. If VPTYPE=D, maximum values are 8000000.
Default:	for BP8K0-BP8K9, 0; for BP16K0-BP16K9, 0;for BP32K, 24; for BP32K1-BP32K9, 0
Update:	ALTER BUFFERPOOL command
DSNZPxxx:	none

Specify the total number or 8KB, 16KB, or 32KB buffers in the given virtual buffer pool (BP8K0-BP8K9, BP16K0-BP16K9, or BP32K-BP32K9).

### 1-30. Hiperpool

Acceptable values:	0-1048576 for 8KB hiperpools, 0-524288 for 16KB 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 (BP8K0-BP8K9, BP16K0-BP16K9, 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. **VPTYPE**

Acceptable values:	P, D
Default:	P
Update:	ALTER BUFFERPOOL command
DSNZPxxx:	none

Specify the type of virtual buffer pool to be allocated.

'D' indicates that the virtual buffer pool is to be allocated in one or more dataspace associated with DB2.

'P' indicates that the virtual buffer pool is to be allocated in the DB2 database services address space.

The VPTYPE specification becomes effective the next time the virtual buffer pool is allocated. If you have already allocated the virtual buffer pool and later change the VPTYPE attribute, the new VPTYPE specification becomes "pending" until the next time the virtual buffer pool is allocated.

# 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 Part 5 (Volume 2) of *DB2 Administration Guide*.

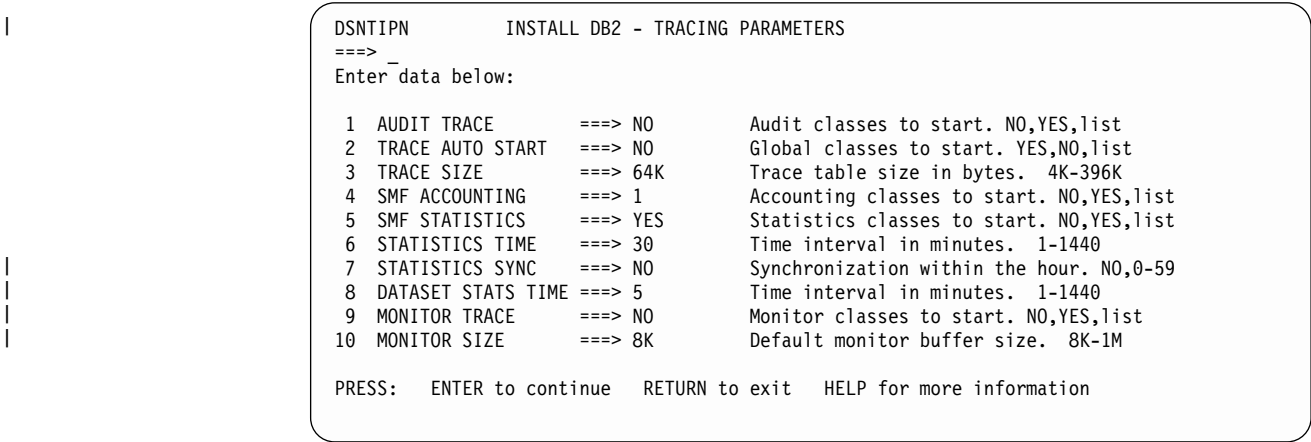


Figure 23. Tracing panel: DSNTIPN

## 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 Part 3 (Volume 1) of *DB2 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 a special START TRACE command as documented in *DB2 Diagnosis Guide and Reference*. 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. For information about the global trace facility, see *DB2 Diagnosis Guide and Reference*.

### 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 Part 5 (Volume 2) of *DB2 Administration Guide*. For information on using the trace, see "Installation step 7: Record DB2 data to SMF (optional)" on page 244.

### 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 Part 5 (Volume 2) of *DB2 Administration Guide*. For information on using the trace, see “Installation step 7: Record DB2 data to SMF (optional)” on page 244.

## 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. STATISTICS SYNC

Acceptable values:	NO, 0-59
Default:	NO
Update:	option 15 on panel DSNTIPB
DSNZPxxx:	DSN6SYSP SYNCVAL

Specify whether DB2 statistics recording is synchronized with some part of the hour. You can specify that the DB2 statistics recording interval is synchronized with the beginning of the hour (0 minutes past the hour) or any number of minutes past the hour up to 59. If NO is specified, no synchronization is done. NO is the default. This parameter has no effect if STATIME is greater than 60.

For example, assume you want the DB2 statistics recording interval to have a length of 15 minutes and to be synchronized with 15 minutes past the hour, which means that DB2 statistics are recorded at 15, 30, 45, and 60 minutes past the hour. To establish this interval, specify the following: STATIME=15, SYNCVAL=15.

## 8. DATASET STATS TIME

Acceptable values:	1-1440
Default:	5 (minutes)
Update:	option 15 on panel DSNTIPB
DSNZPxxx:	DSN6SYSP DSSTIME

The value in this field specifies the time interval, in minutes, between the resetting of data set statistics for online performance monitors. Online performance monitors can request DB2 data set statistics for the current interval with an IFI READS request for IFCID 0199.

## 9. 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 F (Volume 2) of *DB2 Administration Guide*.

## 10. 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).

## 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, NORUN, or 1-5000000
 7 PARAMETER MODULE     ===> DSNZPARM   Name of DB2 subsystem parameter module
 8 AUTO BIND            ===> YES        Use automatic bind. YES, NO, or COEXIST
 9 EXPLAIN PROCESSING   ===> YES        Explain allowed on autobind? YES or NO
10 DPROF SUPPORT        ===> 1          1=NO 2=ONLY 3=ANY
11 SITE TYPE            ===> LOCALSITE   LOCALSITE or RECOVERYSITE
12 TRACKER SITE         ===> NO         Tracker DB2 system. NO or YES
13 READ COPY2 ARCHIVE   ===> NO         Read COPY2 archives first. NO or YES
14 STATISTICS HISTORY   ===> NONE       Default for collection of stats history
15 STATISTICS ROLLUP    ===> NO         Allow statistics aggregation. NO or YES
PRESS:  ENTER to continue  RETURN to exit  HELP for more information
```

Figure 24. 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 Part 5 (Volume 2) of *DB2 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 5000000 is the default limit; if the limit is exceeded, the SQL statement is terminated. For guidelines in choosing the default limit, see Part 5 (Volume 2) of *DB2 Administration Guide*.

#### 7. PARAMETER MODULE

Acceptable values:	1-8 characters
Default:	DSNZPARM

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, COEXIST
Default:	YES
Update:	option 16 on panel DSNTIPB
DSNZPxxx:	DSN6SPRM ABIND

Specify whether plans or packages are automatically rebound (or autobound) at run-time in certain situations. Automatic rebinds can improve availability and administration of plans and packages, because you do not have to explicitly rebind some plans and packages. However, automatic rebinds can also hinder performance because they require access to the DB2 directory and catalog.

Specify the AUTO BIND value as YES, NO or COEXIST where,

### YES

Specifying YES allows automatic rebind operations to be performed at execution time when a plan or package:

- Is "invalid" (the SYSPLAN or SYSPACKAGE column VALID contains 'N', See Part 4 of *DB2 Application Programming and SQL Guide* to see why a plan or package becomes invalid.)
- Was last bound in DB2 Version 7, but is now running on a previous version of DB2.

This autobind is performed on the previous version of DB2 because that is where the user is attempting to run the plan or package. This autobind is performed only for the first attempt to run the plan or package on the previous version of DB2.

- Was last autobound on a previous version of DB2, but is now running again on DB2 Version 7.

This "Version 6 to Version 7" autobind is performed on DB2 Version 7 because that is where the user is attempting to run the plan or package. This autobind is performed only for the first attempt to run the plan/package on Version 7 after an autobind on the previous version. If the plan or package is later run again on the previous version of DB2, the autobind process starts again as previously described.

### NO

Specifying NO for this field prevents DB2 from performing any automatic rebind operations under any circumstances. This means that when the following conditions are true, you must explicitly rebind the plan or package before it can be run again:

- you want to run a plan or package that is "invalid" (the SYSPLAN or SYSPACKAGE column VALID contains 'N', See Part 4 of *DB2 Application Programming and SQL Guide* to see why a plan or package becomes invalid.)
- you are attempting to run a plan or package on a previous version that was last bound (explicitly or automatically) in DB2 Version 7.

- you are attempting to run a plan or package on DB2 Version 7 for which DB2 last performed an autobind. If AUTO BIND = NO has always been in effect on the previous version, then the autobind could not have previously been done on the previous version.

If you attempt to run any plan or package on DB2 Version 7 in one of the situations described above, you will receive SQLCODE -908 SQLSTATE 23510.

## COEXIST

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 Part 4 of *DB2 Application Programming and SQL Guide* to see why a plan or package becomes invalid), or
- was last bound on DB2 Version 7 and is now running on a previous version of DB2.

For this case, DB2 will perform an autobind on the previous version 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 the previous version of DB2.

An automatic rebind operation will not be performed on DB2 Version 7 in a data sharing coexistence environment for any Version 7 plan or package for which DB2 last performed an autobind on the previous version and that plan or package is now run again on Version 7. The plan or package will run on Version 7 as a previous version-bound plan or package, but no Version 7 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 7.

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 "previous version to Version 7" autobind on Version 7 after a previous "Version 7 to a previous version" autobind completed on the previous version.

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 7 and is now run on a previous version of DB2 (a "Version 7 to a previous version" autobind occurs on the previous version), or

- DB2 last performed a "Version 7 to a previous version" autobind for that plan or package on the previous version of DB2, and the plan or package is now run on Version 7 (a "previous version to Version 7" autobind occurs on Version 7).

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 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.

For more information about DataPropagator NonRelational, see DataPropagator NonRelational MVS/ESA Administration Guide.

## 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.

## 12. TRACKER SITE

Acceptable values:	NO or YES
Default:	NO
Update:	option 16 on panel DSNTIPB
DSNZPxxx:	DSN6SPRM TRKRSITE

The value in this field lets you indicate whether the subsystem being installed is to be used as a remote tracker site for another DB2 subsystem to be used in case of a disaster. See Part 4 (Volume 1) of *DB2 Administration Guide* for more information about disaster recovery using a tracker site.

## 13. READ COPY2 ARCHIVE

Acceptable values:	NO or YES
Default:	NO
Update:	option 16 on panel DSNTIPB
DSNZPxxx:	DSN6LOGP ARC2FRST

The value in this field indicates whether COPY2 archives should be read first when the DB2 subsystem is started.

## 14. STATISTICS HISTORY

Acceptable values:	SPACE, NONE, ALL, ACCESSPATH
--------------------	------------------------------



Default: NONE  
Update: option 16 on panel DSNTIPB  
DSNZPxxx: DSN6SPRM STATHIST

Specify which inserts and updates are recorded in catalog history tables.

- SPACE

When the SPACE option is specified, all inserts and updates made by DB2 to space related catalog statistics are recorded.

- ACCESSPATH

When the ACCESSPATH option is specified, all inserts and updates made by DB2 to ACCESSPATH related catalog statistics are recorded.

- ALL

When the ALL option is specified, all inserts and updates made by DB2 in the catalog are recorded.

- NONE

When the NONE option is specified, changes made in the catalog by DB2 are not recorded. This is the default for the HISTORY subsystem parameter.

## 15. STATISTICS ROLLUP

Acceptable values: YES or NO  
Default: NO  
Update: option 16 on panel DSNTIPB  
DSNZPxxx: DSN6SPRM STATROLL

Specify whether the RUNSTATS utility will aggregate the partition level statistics, even though some parts may not contain data. For DB2 systems that have large partitioned table spaces/indexes, the recommended parameter is YES. This will enable the aggregation of part level statistics and will help the optimizer to choose a better access path.

---

## 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.

Most of the 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.

Many of the fields on this panel involve the selection of coded character set identifiers (CCSIDs). Keep the following in mind when choosing values for these fields:

- If you choose YES for the MIXED DATA field, you must specify a Mixed Data CCSID from Table 125 on page 526 or Table 126 on page 526. 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 an incorrect CCSID 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 or need to convert to a CCSID that supports the euro symbol, you can correct it by altering the CCSID field for your default encoding scheme. See “Converting to the euro symbol” on page 528 for the detailed steps. 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 database 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.
- During code conversion, DB2 first looks in the SYSSTRINGS table to see if a conversion is defined. If a conversion is found, it is used. Otherwise, if OS/390 V2R9 is installed, DB2 checks with Language Environment to see if the conversion is supported. See OS/390 C/C++ Programming Guide for additional conversions that might be supported. If the conversion is not available, an error occurs.

For more information on the fields on this panel, see Part 4 of *DB2 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 CCSID         ===> 0        CCSID of your SBCS or MIXED DATA
 9 ASCII CCSID          ===> 0        CCSID of SBCS or mixed data. 0-65533.
10 UNICODE CCSID        ===> 1208     CCSID of UNICODE UTF-8 data.
11 DEF ENCODING SCHEME  ===> EBCDIC   EBCDIC, ASCII, or UNICODE
12 LOCALE LC_CTYPE      ===>
13 APPLICATION ENCODING ===> EBCDIC   EBCDIC, ASCII, UNICODE, ccsid (1-65533)
14 DECIMAL ARITHMETIC   ===> DEC15    DEC15, DEC31, 15, 31
15 USE FOR DYNAMICRULES ===> YES      YES or NO
16 DESCRIBE FOR STATIC  ===> NO       Allow DESCRIBE for STATIC SQL. NO or YES.

PRESS:  ENTER to continue  RETURN to exit  HELP for more information

```

Figure 25. 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 *DB2 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 with DYNAMICRULES run behavior:
  - Whether the value of field DECIMAL POINT IS is COMMA or PERIOD, DB2 recognizes the value as the decimal point for numbers.
- For dynamic SQL statements with DYNAMICRULES bind, define, or invoke behavior:
  - 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 or COMMA, DB2 recognizes the value 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 2 of *DB2 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 37 shows you the different combinations of character string delimiters you get by specifying different values in fields 4 and 5.

*Table 37. Effect of fields 4 and 5 on SQL and COBOL string delimiters*

**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 38 shows you why you might specify different combinations of values in these fields.

*Table 38. 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

The MIXED DATA option indicates how the EBCDIC CCSID and ASCII CCSID fields are to be interpreted by DB2. The MIXED DATA option has no affect on UNICODE CCSID field. Regardless of the setting for MIXED DATA, UNICODE UTF-8 data will be considered mixed data and will be processed according to the rules for mixed data.

With MIXED DATA YES, the CCSID specified in either the EBCDIC CCSID or ASCII CCSID field must be the mixed CCSID for the encoding scheme. From this, DB2 will determine the associated SBCS and DBCS CCSIDs for the encoding scheme. MIXED DATA YES allows EBCDIC and ASCII mixed character data and graphic data to be defined.

With MIXED DATA NO, the CCSID specified in either the EBCDIC CCSID or ASCII CCSID field must be the CCSID for the encoding scheme. MIXED DATA NO does not allow for EBCDIC or ASCII mixed character data or graphic data to be defined.

For EBCDIC data, 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 CCSID

Acceptable values:	0-65533
Default:	0 (no CCSID is used)
Update:	not recommended – data integrity may be compromised
DSNHDECP:	SCCSID (single-byte), MCCSID (mixed), GCCSID (graphic)

If your installation does not use character conversion, accept the default. You must specify a coded character set identifier (CCSID) if you have any EBCDIC or ASCII mixed character data, graphic data, or 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 124 on page 524.

To select a CCSID for mixed-data (an MCCSID), see Table 125 on page 526. By specifying a CCSID you also receive system CCSIDs for your SBCS and GRAPHIC data.

If you specify either 930 or 5026, Katakana characters are allowed in ordinary identifiers and letters are not changed to upper case.

## 9. ASCII CCSID

Acceptable values:	0-65533
Default:	0 (no CCSID is used)

Update:	not recommended – data integrity may be compromised
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.

To determine which single-byte ASCII CCSID to use, see Table 124 on page 524. To determine which double-byte ASCII CCSID to use, see Table 126 on page 526. See also the *Character Data Representation Architecture Overview* and *Character Data Representation Architecture Reference and Registry* for more information.

## 10. UNICODE CCSID

Acceptable values:	1208
Default:	1208
Update:	Not Recommended – data integrity may be compromised
DSNHDECP:	USCCSID (367 for single-byte), UMCCSID (1208 for mixed), UGCCSID (1200 for graphic)

Accept the default CCSID for Unicode. DB2 currently only allows specification of CCSID 1208 for this value. DB2 will automatically choose the CCSIDs for double-byte and single-byte data. Do not change CCSID values once they have been specified. SQL results may be unpredictable if this recommendation is not followed.

## 11. DEF ENCODING SCHEME

Acceptable values:	EBCDIC, ASCII, UNICODE
Default:	EBCDIC
Update:	not recommended
DSNHDECP:	ENSCHHEME

Specify the default format in which to store data in DB2. If you specify DEF ENCODING SCHEME=ASCII or EBCDIC and MIXED DATA=YES, specify a mixed CCSID.

## 12. LOCALE LC\_CTYPE

Acceptable values:	A valid locale of 0-50 characters. See <i>OS/390 C/C++ Programming Guide</i> or <i>DB2 SQL Reference</i> for the format of valid LOCALE LC_CTYPE names.
Default:	Blank
Update:	option 17 of DSNTIPB
DSNHDECP:	LC_TYPE

Specify the system LOCALE LC\_CTYPE. A *locale* is the part of your system environment that depends on language and cultural conventions. An LC\_TYPE is a subset of a locale that applies to character functions.

The UPPER, LOWER, and TRANSLATE scalar functions use the CURRENT LOCALE LC\_CTYPE system default or special register. The results of these functions can vary, depending on the setting of the locale.

**Recommendation:** Use the default value for LOCALE LC\_CTYPE unless you need to execute the UPPER, LOWER, or TRANSLATE functions for data that must be interpreted using the rules provided by specific locales.

**Examples:** En\_US or Fr\_CA.

### 13. APPLICATION ENCODING

Acceptable values:	ASCII, EBCDIC, UNICODE, or ccsid (between 1- 65533) 7 characters)
Default:	EBCDIC
Update:	Not Recommended
DSNHDECP:	APPENSCH

Specify the system default application encoding scheme.

The system default application encoding scheme affects how DB2 interprets input and output data. For example, if you set your default application encoding scheme to 37, and your EBCDIC coded character set to 500, then DB2 will convert all input and output data to 500 from 37 before using it. This includes, but is not limited to, SQL statement text and host variables.

**Important:** If you change the value of this field to an encoding scheme other than EBCDIC, you may experience a release incompatibility. The default value, EBCDIC, will cause DB2 to retain the behavior of previous versions.

### 14. 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 by becoming the initial value for the CURRENT PRECISION special register and it 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 2 of *DB2 SQL Reference* for information about arithmetic with two decimal operands.

### 15. USE FOR DYNAMICRULES

Acceptable values:	YES or NO
Default:	YES
Update:	option 17 on panel DSNTIPB
DSNHDECP:	DYNRULS

Specify whether DB2 should use the application programming defaults specified on this panel or use the values of the DB2 precompiler options for dynamic SQL statements bound using DYNAMICRULES bind, define, or invoke behavior.

Specify YES to use the application programming defaults for these fields regardless of the DYNAMICRULES option:

- DECIMAL POINT IS
- STRING DELIMITER



- SQL STRING DELIMITER
- MIXED DATA
- DECIMAL ARITHMETIC

Specify NO to use the DB2 precompiler values for dynamic SQL statements in plans or packages bound using the DYNAMICRULES bind, define, or invoke behavior.

## 16. DESCRIBE FOR STATIC

Acceptable values:	NO or YES
Default:	NO
Update:	option 17 on panel DSNTIPB
DSNZPxxx:	DSN6SPRM DESCSTAT

This option controls whether DB2 builds a DESCRIBE SQLDA when binding static SQL statements. Normally, a DESCRIBE cannot be issued against a static SQL statement, with the following exceptions:

- In a distributed environment, where DB2 for OS/390 and z/OS is the server and the requester supports extended dynamic SQL. In this scenario, a DESCRIBE request that is executed on an SQL statement in the extended dynamic package appears to DB2 as a DESCRIBE on a static SQL statement in the DB2 package.
- When an application uses a stored procedure result set, the application must allocate a cursor for that result set. The application can describe that cursor using a DESCRIBE CURSOR statement. The SQL statement that is actually described is the one with the cursor declared in the stored procedure. If that statement is static, then this requires that a static SQL statement be described.

NO, the default means that DB2 does not generate a DESCRIBE SQLDA at BIND time for static SQL statements. If a DESCRIBE request is received at execution time, DB2 generates an error. However, if the describe request comes from a DESCRIBE CURSOR statement, DB2 satisfies the request but is only able to provide data type and length information. Column names are not provided.

YES means that DB2 does generate a DESCRIBE SQLDA at BIND time so that DESCRIBE requests for static SQL can be satisfied during execution. You must rebind the package after this value has been set to YES. Specifying YES increases the size of some packages because the DESCRIBE SQLDA is now stored with each statically-bound SQL SELECT statement.

## 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).

For more information on the fields in this panel, see Chapter 2 of *DB2 SQL Reference*.

```
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 CURRENT DEGREE       ===> 1        1 or ANY
 7 CACHE DYNAMIC SQL    ===> NO       NO or YES
 8 OPTIMIZATION HINTS   ===> NO       Enable optimization hints. NO or YES
 9 VARCHAR FROM INDEX   ===> NO       Get VARCHAR data from index. NO or YES
10 RELEASE LOCKS        ===> YES      Release cursor with hold locks. YES,NO
11 MAX DEGREE           ===> 0        Maximum degree of parallelism. 0-254
12 UPDATE PART KEY COLS ===> YES      Allow update of partitioning key
                                   columns. YES, NO, SAME
13 LARGE EDM BETTER FIT ===> NO       NO or YES
14 IMMEDIATE WRITE      ===> NO       NO, YES or PH1
15 EVALUATE UNCOMMITTED ===> NO       Evaluate uncommitted data. NO or YES

PRESS:  ENTER to continue  RETURN to exit  HELP for more information
```

Figure 26. 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 39 as a default output format to represent dates:

Table 39. Date formats

Format Name	Abbreviation	Format	Example
International Standards Organization	ISO	yyyy-mm-dd	1995-12-23
IBM USA standard	USA	mm/dd/yyyy	12/23/1995
IBM European standard	EUR	dd.mm.yyyy	23.12.1995
Japanese Industrial Standard Christian Era	JIS	yyyy-mm-dd	1995-12-23
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 *DB2 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 40 as a default output to represent times:

Table 40. Time formats

Format Name	Abbreviation	Format	Example
International Standards Organization	ISO	hh.mm.ss	13.30.05
IBM USA standard	USA	hh:mm AM or hh PM	1:30 PM or 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 *DB2 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 1 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 2 on this panel.

## 5. STD SQL LANGUAGE

Acceptable values:	YES, NO
Default:	NO
Update:	option 18 on panel DSNTIPB
DSNHDECP:	STDSQL

To specify that the SQL language used in application programs conforms to the portions of the 1992 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. 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.

## 7. 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 45 for storage estimating details.

## 8. OPTIMIZATION HINTS

Acceptable values:	NO or YES
Default:	NO
Update:	option 18 on panel DSNTIPB
DSNZPxxx:	DSN6SPRM OPTHINTS

The value in this field allows you to pass information to DB2 that might influence the access path selected for certain queries. The information that is passed is in the form of rows in the PLAN\_TABLE. See Part 5 (Volume 2) of *DB2 Administration Guide* for more information on the PLAN\_TABLE.

## 9. VARCHAR FROM INDEX

Acceptable values:	NO or YES
Default:	NO
Update:	option 18 on panel DSNTIPB
DSNZPxxx:	DSN6SPRM RETVLCFK

Specify whether the VARCHAR column will be retrieved from the index. The data sharing scope of this parameter is GROUP.

If you choose NO, DB2 must go to the data page to retrieve data because index-only access of varying-length column data will not be possible. With a setting of NO, data is retrieved with no padding.

If you choose YES, better performance will result because index-only access of varying-length column data will be enabled. The data retrieved from the index is padded with blanks to the maximum length of the column.

**Attention:** Applications must be able to handle the padding blanks. If your application is sensitive to these blanks, keep the default value of NO. You must rebind plans and packages to enable the change.

## 10. RELEASE LOCKS

Acceptable values:	YES or NO
Default:	YES
Update:	option 18 on panel DSNTIPB
DSNZPxxx:	DSN6SPRM RELCURHL

Specify whether DB2 should, at commit time, release a data page or row lock on which a cursor defined WITH HOLD is positioned. This lock is *not* necessary for maintaining cursor position. See Part 5 (Volume 2) of *DB2 Administration Guide* for more information about locks for cursors defined WITH HOLD.

If you choose YES, the default, DB2 releases this data page or row lock after a COMMIT is issued for cursors defined WITH HOLD. Specifying YES can improve concurrency.

If you choose NO, DB2 holds the data page or row lock for WITH HOLD cursors after the COMMIT. This option is provided so that existing applications that rely on this lock can continue to work correctly.

## 11. MAX DEGREE

Acceptable values:	0-254
Default:	0
Update:	option 18 on panel DSNTIPB
DSNZPxxx:	DSN6SPRM PARAMDEG

Specify the maximum degree of parallelism for a parallel group if non-zero. When you specify a value, you limit the degree of parallelism so that DB2 can not create too many parallel tasks which use virtual storage. The default value of zero means that DB2 will choose a maximum degree of parallelism that is based on the system configuration. See Part 5 (Volume 2) of *DB2 Administration Guide* for more information about limiting the degree of parallelism.

## 12. UPDATE PART KEY COLS

Acceptable values:	YES, NO, or SAME
Default:	YES
Update:	option 18 on panel DSNTIPB
DSNZPxxx:	DSN6SPRM PARTKEYU

#  
#

Specify whether values in columns that participate in partitioning keys may be updated. The acceptable values mean the following:

- YES** Values in columns that participate in partitioning keys may be updated. This is the default.
- NO** Values in columns that participate in partitioning keys may not be updated.
- SAME** Values in columns that participate in partitioning keys may be updated if and only if the update leaves the row belonging to the same partition.

Attempts to inappropriately update the value in a partitioning key column result in the failure of the SQL statement with a -904 resource unavailable SQL code. The accompanying DSNT501I message identifies the unavailable resource as code (type X'3000'), and uses reason code of 00C90097.

### 13. LARGE EDM BETTER FIT

Acceptable values: YES, NO  
 Default: NO  
 Update: option 18 on panel DSNTIPB  
 DSNZPxxx: DSN6SPRM EDMBFIT

Specify how free space is utilized for large EDM pools (greater than 40M). Specify NO to optimize for performance. YES optimizes for better storage utilization. In the trade-off between performance and space utilization, space is normally more critical for smaller EDM pools and performance is more critical for larger EDM pools.

### 14. IMMEDIATE WRITE

Acceptable values: YES, NO, PH1  
 Default: NO  
 Update: option 18 on panel DSNTIPB  
 DSNZPxxx: DSN6GRP IMMEDWRI

Specify when updates to group buffer pool-dependent buffers are to be written to the coupling facility (CF). If you specify NO, DB2 will not immediately write the change buffer to the CF. Instead, it will wait until commit. If you specify YES, it will immediately write the page to the CF after it is updated. If you specify PH1, DB2 will write the updated page during phase 1 commit.

Table 41 illustrates the implied heirarchy when using the IMMEDWRI subsystem parameter and the IMMEDIATE WRITE option of the BIND and REBIND commands.

*Table 41. The implied hierarchy of the immediate write option*

<b>IMMEDIATE</b>	<b>IMMEDIATE</b>	
<b>bind option</b>	<b>subsystem parameter</b>	<b>Value at 'run time'</b>
NO	NO	NO
NO	PH1	PH1
NO	YES	YES
PH1	NO	PH1
PH1	PH1	PH1
PH1	YES	YES
YES	NO	YES
YES	PH1	YES
YES	YES	YES

## 15. EVALUATE UNCOMMITTED

Acceptable values:	NO, YES
Default:	NO
DSNZPxxx:	DSN6SPRM EVALUNC

Specify whether predicate evaluation can occur on uncommitted data of other transactions. The option applies only to stage 1 predicate processing that uses table access (table space scan, index-to-data access, and RID list processing) for queries with isolation level RS or CS.

Although the option influences whether predicate evaluation can occur on uncommitted data, it does not influence whether uncommitted data is returned to an application. Queries with isolation level RS or CS will return only committed data. They will never return the uncommitted data of other transactions, even if predicate evaluation occurs on such. If data satisfies the predicate during evaluation, the data is locked as needed, and the predicate is re-evaluated as needed before the data is returned to the application.

If you specify NO, the default, predicate evaluation occurs only on committed data (or on the application's own uncommitted changes). NO ensures that all qualifying data is always included in the answer set.

If you specify YES, predicate evaluation can occur on uncommitted data of other transactions. With YES, data can be excluded from the answer set. Data that does not satisfy the predicate during evaluation but then, because of undo processing (ROLLBACK or statement failure), reverts to a state that does satisfy the predicate is missing from the answer set. A value of YES enables DB2 to take fewer locks during query processing. The number of locks avoided depends on:

- the query's access path
- the number of evaluated rows that do not satisfy the predicate
- the number of those rows that are on overflow pages

**Recommendation:** Specify YES to improve concurrency if your applications can tolerate returned data to falsely exclude any data that would be included as the result of undo processing (ROLLBACK or statement failure).

## 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 36 for more information about DB2 and IRLM. See *DB2 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
 9  X LOCK FOR SEARCHED U/D ====> NO  Use X lock for serached updates or
                                         deletes. NO or YES.
10  START IRLM CTRACE ====> NO       Start IRLM component traces at startup?
                                         NO or YES
11  IMS BMP TIMEOUT   ====> 4        Timeout multiplier for BMP. 1-254
12  DL/I BATCH TIMEOUT ====> 6       Timeout multiplier for DL/I. 1-254
13  RETAINED LOCK TIMEOUT ====> 0    Retained lock timeout multiplier. 0-254
PRESS:  ENTER to continue  RETURN to exit  HELP for more information
```

Figure 27. IRLM panel 1: DSNTIPI

### 1. INSTALL IRLM

Acceptable values:	YES, NO
Default:	YES
Update:	see note B on page 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.



If you do not have a new procedure created, ensure that your old IRLM procedure gets updated with any new keywords that were added.

## 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 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 188.

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 Version 2 installed, use the MVS subsystem name for that IRLM. Otherwise, accept the default value, IRLM. For more information, see *IMS Command Reference*.

IRLM PROC parameter: IRLNMN

## 3. RESOURCE TIMEOUT

Acceptable values:	1-3600
Default:	60
Update:	see note B on page 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. This value is rarely the actual time. For data sharing, the actual timeout period is longer than the timeout value. See Chapter 7 of *DB2 Data Sharing: Planning and Administration* for an explanation of timeouts.

For information about optimizing performance by managing DB2's use of locks, see Part 5 (Volume 2) of *DB2 Administration Guide*.

## 4. AUTO START

Acceptable values:	YES, NO
Default:	YES
Update:	option 19 on panel DSNTIPB
DSNZPxxx:	DSN6SPRM IRLMAUT

Specify whether DB2 automatically starts and stops the IRLM.

If you specify YES, when DB2 starts, it tries to start the IRLM if the IRLM is not already started. When DB2 stops, IRLM automatically stops if the IRLM was started by DB2. However, IRLM will not automatically terminate if it is defined with "DISCONNECT IRLM = NO" option. **Recommendation:** Use 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.

When IRLM initializes, it is registered with the MVS automatic restart manager (ARM). It deregisters from the ARM when the IRLM is shut down normally. When IRLM terminates, it sends DB2 the registration information so that DB2 can determine whether IRLM was terminated normally. DB2 then deregisters from the ARM to prevent unwanted restarts. Otherwise, IRLM might be restarted with DB2, if AUTO START is YES. See Part 4 (Volume 1) of *DB2 Administration Guide* for information on changing the IRLM policy definition in a non-data sharing environment. See *DB2 Data Sharing: Planning and Administration* for IRLM policy definitions within a data sharing environment.

## 5. PROC NAME

Acceptable values:	1-8 characters
Default:	IRLMPROC
Update:	see note B on page 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 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 Part 5 (Volume 2) of *DB2 Administration Guide*.

## 8. U LOCK FOR RR/RS:

Acceptable values:	YES, NO
Default:	NO
Update:	option 19 on panel DSNTIPB
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.

#### 9. X LOCK FOR SEARCHED U/D:

Acceptable values:	YES, NO
Default:	NO
Update:	option 19 on panel DSNTIPB
DSNZPxxx:	DSN6SPRM XLKUPDLT

Specify 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. Use this option to achieve higher rates of concurrency.

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 value of YES is beneficial in a data sharing environment when most or all searched updates and deletes use an index. If YES is specified and searched updates or deletes result in a table space scan, the likelihood of timeouts and deadlocks greatly increases.

#### 10. START IRLM CTRACE

Acceptable values:	NO or YES
Default:	NO
Update:	option 19 on panel DSNTIPB
DSNZPxxx:	none

Specify whether the IRLM component traces should be activated when IRLM is started. The DB2-provided IRLM proc in DSNTIJMV is tailored according to this value.

Specify No if you want IRLM to start only the low-activity subtraces EXP, INT, and XIT.

Specify YES if you want IRLM to start with all its subtraces active. Starting all IRLM subtraces slightly impacts performance demand for ECSA storage, but improves serviceability.

#### 11. IMS BMP TIMEOUT

Acceptable values:	1-254
Default:	4
Update:	option 19 on panel DSNTIPB

DSNZPxxx:

DSN6SPRM BMPTOUT

Specify the number of resource timeout values (field 3) that an IMS BMP connection waits for a lock to be released. For example, if you use the default value, an IMS BMP connection can wait 4 times the resource timeout value for a resource. This option gives you flexibility in tuning your system to avoid timeouts. For more information, see topic Installation options for wait times in Part 5 (Volume 2) of *DB2 Administration Guide*.

In Version 5, acceptable values were 0-254, with a 0 meaning to use the default value of 4.

## 12. DL/I BATCH TIMEOUT

Acceptable values: 1-254

Default: 6

Update: option 19 on panel DSNTIPB

DSNZPxxx: DSN6SPRM DLITOUT

Specify the number of resource timeout values (field 3) that a DL/I batch connection waits for a lock to be released. For example, if you use the default value, a DL/I batch application can wait 6 times the resource timeout value for a resource. This option gives you flexibility in tuning your system to avoid timeouts. For more information, see topic Installation options for wait times in Part 5 (Volume 2) of *DB2 Administration Guide*.

In Version 5, acceptable values were 0-254, with a 0 meaning to use the default value of 6.

## 13. RETAINED LOCK TIMEOUT

Acceptable values: 0-254

Default: 0

Update: option 19 on panel DSNTIPB

DSNZPxxx: DSN6SPRM RETLWAIT

This value is of importance only in a data sharing environment. This value lets you indicate how long a transaction should wait for a lock on a resource if another DB2 in the data sharing group has failed and is holding an incompatible lock on that resource. Locks held by failed DB2 members are called *retained locks*. For more information, see *DB2 Data Sharing: Planning and Administration*.

The value you use is a multiplier that is applied to the connection's normal timeout value. For example, if the retained lock multiplier is 2, then the timeout period for a call attachment connection that is waiting for a retained lock is  $1 \times 2$  (1 for the normal CAF timeout period, 2 for the additional time specified for retained locks).

If you use the default, 0, then applications do not wait for incompatible retained locks, but instead the lock request is immediately rejected and the application receives a "resource unavailable" SQLCODE.

In Version 5, acceptable values were YES and NO. YES corresponds to a 1, and NO corresponds to a 0.

## 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

PAGE PROTECT

===> YES

Page Protect Common Modules 3 MAXIMUM ECSA

4

LOCKS PER TABLE(SPACE)

===> 1000

Maximum before lock escalation

5

LOCKS PER USER

===> 10000

Maximum before resource unavailable

6

DEADLOCK TIME

===> 5

Detection interval (1-5 seconds or 100-5000 milliseconds)

For DB2 data sharing ONLY enter data below:

7

DEADLOCK CYCLE

===> 1

Number of LOCAL cycles before GLOBAL

8

MEMBER IDENTIFIER

===> 1

Member ID for this IRLM (1-255)

9

IRLM XCF GROUP NAME

===> DXRGROUP

Name of IRLM XCF group

10

LOCK ENTRY SIZE

===> 2

Initial allocation, in bytes (2,4,8)

11

NUMBER OF LOCK ENTRIES

===> 0

Lock table entries 12 DISCONNECT IRLM

===> YES

Disc

PRESS: ENTER to continue RETURN to exit HELP for more information

Figure 28. 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 for the lock control block structure.

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. IRLM still uses CSA and ECSA for other purposes. With PC=YES, the MAXIMUM ECSA option is ignored. See “Common service area” on page 37 for more information about how DB2 uses this field.

IRLM PROC parameter: PC

#  
  
#  
#  
#  
#

### 2. PAGE PROTECT

Acceptable values: NO, YES  
Default: YES  
Update: edit IRLM start procedure  
DSNZPxxx: none

#  
# Specify whether IRLM loads its common storage modules into page protected  
# storage.

YES, the default, indicates that modules located in common storage are to be loaded into page protected storage to prevent programs from overlaying the instructions. YES is recommended because there is no additional overhead once the modules are loaded, and the protection can prevent code overlay failures.

#  
# NO indicates that common storage modules will be loaded into CSA or ECSA  
# without first page protecting that memory.

# IRLM PROC parameter: PGPROT

### 3. 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 put a larger value than you think you might need in here. You can also change the value dynamically by using the MVS command `MODIFY irImproc,SET,CSA`. See "Common service area" on page 37 for more information about how IRLM uses storage.

IRLM PROC parameter: MAXCSA

### 4. 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 Part 5 (Volume 2) of *DB2 Administration Guide*.

### 5. LOCKS PER USER

Acceptable values:	0-100000
Default:	10000
Update:	option 20 on panel DSNTIPB
DSNZPxxx:	DSN6SPRM NUMLKUS

Specify the maximum number of page, row, or LOB 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, 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 Part 5 (Volume 2) of *DB2 Administration Guide*.

## 6. DEADLOCK TIME

```
# Acceptable values:      1-5, 100-5000
# Default:               5
# Update:                edit IRLM start procedure
# DSNZPxxx:              none
#
```

Specify the time, in seconds or milliseconds, of the local deadlock detection cycle. Depending on the value that you enter, IRLM might substitute a smaller maximum value. 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. If you specify a value between 1 and 5, the time is assumed to be in seconds; if you specify a value between 100 and 5000, the time is assumed to be in milliseconds.

IRLM PROC parameter: DEADLOK

## 7. DEADLOCK CYCLE

```
Acceptable values:      1
Default:                1
Update:                 edit IRLM start procedure
DSNZPxxx:               none
```

This option is used only for DB2 data sharing. The value is the number of local deadlock cycles that must expire before the IRLM performs global deadlock detection processing.

IRLM PROC parameter: DEADLOK

## 8. MEMBER IDENTIFIER

```
Acceptable values:      1-255
Default:                1
Update:                 edit IRLM start procedure
DSNZPxxx:               none
```

Specify an ID number that uniquely names this IRLM member within an IRLM data sharing group. **Recommendation:** 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 *DB2 Command Reference* for the *irlmproc* command information.

This option is used only for DB2 data sharing.

IRLM PROC parameter: IRLMID

9. 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. **Recommendation:** Begin this name 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 DXR. Do not use SYS as the first three characters, and do not use UNDESIG as the group name.

IRLM PROC parameter: IRLMGRP

10. 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 42.

Table 42. Converting lock entry size to maxusrs values

LOCK ENTRY SIZE	MAXUSRS value
2	7
4	23
8	32

11. NUMBER OF LOCK ENTRIES

Acceptable values: 1-1024  
Default: 0  
Update: edit IRLM start procedure  
DSNZPxxx: none

Specify the number of lock table entries you want in the coupling facility lock structure. This value must be a power of two in the range of 1 to 1 024 with each increment representing 1 048 576 lock table entries. The default, zero, indicates that IRLM will determine the number of lock table entries. If you wish to control the number of lock table entries, enter a non-zero value within the accepted range. This number has a direct affect on cross-system extended services (XES) contention and should be monitored to find the optimum values for your installation.

IRLM PROC parameter: LTE



## 12. DISCONNECT IRLM

Acceptable values:	YES, NO
Default:	YES
Update:	edit IRLM start procedure
DSNZPxxx:	none

Specify whether IRLM should disconnect automatically from the data sharing group when DB2 is not identified to it. The default, YES, causes IRLM to disconnect from the data sharing group when DB2 is stopped normally or as the result of a DB2 failure.

When you specify YES in conjunction with AUTOSTART YES (on panel DSNTIPI), there is no manual intervention required to stop IRLM.

If you specify NO, IRLM remains connected to the data sharing group even when DB2 is stopped. You must explicitly stop IRLM to bring it down.

With NODISCON, there is less impact on other systems when a DB2 fails because MVS is not required to perform certain recovery actions that it normally performs when IRLM comes down. NODISCON can also mean that DB2 restarts more quickly after a DB2 normal or abnormal termination because it does not have to wait for IRLM to rejoin the IRLM data sharing group.

IRLM PROC parameter: SCOPE

## Protection panel: DSNTIPP

The entries on this panel are used for security matters. Data sets should be protected by a security subsystem, such as RACF, rather than by passwords. Support for password protection was removed in Version 6.

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 and z/OS Version 7 uses your Version 6 or Version 5 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 services. 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 90.

```
DSNTIPP          INSTALL DB2 - PROTECTION
===> _

Enter data below:

 1  ARCHIVE LOG RACF  ===> NO          RACF protect archive log data sets
 2  USE PROTECTION   ===> YES         DB2 authorization enabled. YES or NO
 3  SYSTEM ADMIN 1   ===> SYSADM      Authid of system administrator
 4  SYSTEM ADMIN 2   ===> SYSADM      Authid of system administrator
 5  SYSTEM OPERATOR 1 ===> SYSOPR     Authid of system operator
 6  SYSTEM OPERATOR 2 ===> SYSOPR     Authid of system operator
 7  UNKNOWN AUTHID   ===> IBMUSER     Authid of default (unknown) user
 8  RESOURCE AUTHID  ===> SYSIBM      Authid of Resource Limit Table creator
 9  BIND NEW PACKAGE ===> BINDADD     Authority required: BINDADD or BIND
10  PLAN AUTH CACHE  ===> 1024        Size in bytes per plan (0 - 4096)
11  PACKAGE AUTH CACHE===> 32K        Global - size in bytes (0-2M)
12  ROUTINE AUTH CACHE===> 32K        Global - size in bytes (0-2M)
13  DBADM CREATE AUTH ===> NO         DBA can create views/aliases for others.

PRESS:  ENTER to continue  RETURN to exit  HELP for more information
```

Figure 29. Protection panel: DSNTIPP

### 1. ARCHIVE LOG RACF

Acceptable values:	YES, NO
Default:	NO
Update:	see page 184; 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 Part 3 (Volume 1) of *DB2 Administration Guide*.

## 2. 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.

## 3. 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 Part 3 (Volume 1) of *DB2 Administration Guide*.

## 4. 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.

## 5. 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 Part 3 (Volume 1) of *DB2 Administration Guide*.

If blanked, the value is set to the value of field 8.

## 6. 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.

## 7. 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.

## 8. 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.

## 9. 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 Part 3 (Volume 1) of *DB2 Administration Guide* for a full description of the privileges needed to bind a new package.

## 10. 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 Part 3 (Volume 1) of *DB2 Administration Guide* for more information about cache size.

## 11. 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. 32KB is enough storage for about 375 collection-ID.package-IDs. The cache is stored in the DSN1DBM1 address space.

## 12. ROUTINE AUTH CACHE

Acceptable values:	0-2M
Default:	32K
Update:	option 21 on panel DSNTIPB
DSNZPxxx:	DSN6SPRM CACHERAC

Specify how much storage to allocate for the caching of routine authorization information for all routines on this DB2 member. Routines include stored procedures and user-defined functions. 32K bytes is enough storage for about 380 schema.routine.type entries.

## 13. DBADM CREATE AUTH

Acceptable values:	NO, YES
Default:	NO
Update:	option 21 on panel DSNTIPB
DSNZPxxx:	DSN6SPRM DBACRVW

Specify whether an authorization ID with DBADM authority on a database can:

- create a view for another authorization ID on that database's tables
- create an alias for itself or another authorization ID for a table in that database

If you specify YES, an authorization ID with DBCTRL authority on a database can also create an alias for itself or another authorization ID for a table in that database.

---

## 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
- LNKSTxx, 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, it is recommended 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

2

COMMAND PREFIX

3

SUBSYSTEM MEMBER

4

SUBSYSTEM SEQUENCE

5

AUTH MEMBER

6

AUTH SEQUENCE

7

LINK LIST ENTRY

8

LINK LIST SEQUENCE

9

COMMAND SCOPE

10

SUPPRESS SOFT ERRORS

====>

DSN1

-DSN1

00

88888888

00

88888888

00

88888888

00

88888888

STARTED

YES

Name for connecting to DB2

DB2 subsystem command prefix

xx in IEFSSNxx

Sequence number for insertion

xx in IEAAPFxx APF member name

Sequence number for insertion

xx in LNKSTxx for DSNLINK

Sequence number for insertion

SYSTEM, SYSPLEX, or STARTED

Suppress logrec recording. Yes or No

PRESS:

ENTER to continue

RETURN to exit

HELP for more information

Figure 30. 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

DSNHDECP: SSID

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 188

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 43. The remaining characters of the command prefix must be from Table 43, A-Z, or 0-9.

*Table 43. Allowable special characters for the command prefix*

<b>Name</b>	<b>Character</b>	<b>Hexadecimal Representation</b>
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 236.

### 3. SUBSYSTEM MEMBER

Acceptable values:	2 alphanumeric characters
Default:	00
Update:	see page 188
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 188
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 188
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 because job DSNTIJMV does not do it for you.

### 6. AUTH SEQUENCE

Acceptable values:	1-99999995
Default:	88888888
Update:	see page 188
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 188
DSNZPxxx:	none

Specify the last two characters of LNKLISTxx as needed to include the *prefix*.SDSNLINK library.

## 8. LINK LIST SEQUENCE

Acceptable values:	1-99999999
Default:	88888888
Update:	see page 188
DSNZPxxx:	none

Specify any number greater than the highest sequence number already used in the LNKLISTxx PARMLIB member.

## 9. COMMAND SCOPE

Acceptable values:	SYSTEM, SYSPLEX, STARTED
Default:	STARTED
DSNZPxxx:	none

Specify the scope of the command prefix.

<b>SYSTEM</b>	The scope of commands is for one MVS system. The command prefix is registered at MVS IPL.
<b>SYSPLEX</b>	The scope of commands is for the entire Sysplex. The command prefix is registered at MVS IPL.
<b>STARTED</b>	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. Use STARTED if you intend to use MVS's automatic restart manager, or if you might move this DB2 into a data sharing group.

## 10. SUPPRESS SOFT ERRORS

Acceptable values:	Yes, No
Default:	Yes
Update:	option 22 on panel DSNTIPB
DSNZPxxx:	SUPPERRS

Specify whether to record errors such as invalid decimal data and arithmetic exceptions. DB2 catches these errors and issues SQLCODEs for them. This option enables or disables the recording of these errors in the operating system dataset, SYS1.LOGREC.

## 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 Part 4 (Volume 1) of *DB2 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  OUTPUT BUFFER      ===> 4000K       Size in bytes (40K-400000K)
 3  ARCHIVE LOG FREQ    ===> 24        Hours per archive run
 4  UPDATE RATE        ===> 3600       Updates, inserts, and deletes per hour
 5  LOG APPLY STORAGE   ===> 0M        Maximum ssnmDBM1 storage in MB for
                                       fast log apply (0-100M)

 6  CHECKPOINT FREQ     ===> 50000      Log records or minutes per checkpoint
 7  FREQUENCY TYPE      ===> LOGRECS    CHECKPOINT FREQ units. LOGRECS, MINUTES
 8  UR CHECK FREQ       ===> 0         Checkpoints to enable UR check. 0-255
 9  UR LOG WRITE CHECK  ===> 0K        Log Writes to enable UR check. 0-1000K
10  LIMIT BACKOUT       ===> AUTO      Limit backout processing. AUTO, YES, NO
11  BACKOUT DURATION    ===> 5         Checkpoints processed during backout if
                                       LIMIT BACKOUT = AUTO or YES. 0-255.
12  RO SWITCH CHKPTS    ===> 5         Checkpoints to read-only switch. 1-32767
13  RO SWITCH TIME      ===> 10        Minutes to read-only switch. 1-32767
14  LEVELID UPDATE FREQ===> 5         Checkpoints between updates. 0-32767
PRESS:  ENTER to continue  RETURN to exit  HELP for more information
```

Figure 31. 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 use the DSNJU003 utility to modify the number of logs, your modified value is not reflected on this panel. See "Updating other parameters" on page 233 for details.

### 2. OUTPUT BUFFER

Acceptable values: 40K-400000K  
Default: 4000K  
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.

### 3. 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.

#### 4. 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.

#### 5. LOG APPLY STORAGE

Acceptable values:	0-100M
Default:	0M
Update:	option 23 on panel DSNTIPB
DSNZPxxx:	DSN6SYSP LOGAPSTG

The value in this field represents the maximum *ssnmDBM1* storage per member that can be used by the fast log apply process. The default value is 0 MB, which means that fast log apply is disabled except during DB2 restart. During DB2 restart, fast log apply is always enabled because there are no DB2 applications running. When using fast log apply, be sure that the storage value you specify is not needed by DB2 application programs.

#### 6. CHECKPOINT FREQ

Acceptable values:	200-16 000 000 (log records), or 1-60 (minutes)
Default:	50 000
Update:	option 23 on panel DSNTIPB
DSNZPxxx:	DSN6SYSP CHKFREQ

Specify the system checkpoint frequency in minutes or in number of log records. If you have widely variable logging rates, maximize system performance by specifying the checkpoint frequency in time. DB2 will start a new checkpoint at the interval you specify, either in minutes, or number of log records.

The SET LOG command can be used to dynamically change the number of log records between checkpoints.

## 7. FREQUENCY TYPE

Acceptable values:	LOGRECS or MINUTES
Default:	LOGRECS
Update:	cannot change during update
DSNZPxxx:	none

Specify whether the units for checkpoint frequency are in log records or time. If you choose LOGRECS, specify the number of records in the CHECKPOINT FREQ field. If you choose MINUTES, specify the number of minutes.

## 8. UR CHECK FREQ

Acceptable values:	0-255
Default:	0
Update:	option 23 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.

## 9. UR LOG WRITE CHECK

Acceptable values:	0-1000K
Default:	0
Update:	option 23 on panel DSNTIPB
DSNZPxxx:	DSN6SYSP URLGWTH

Specify the number of log records that are written by an uncommitted unit of recovery (UR) before DB2 will issue a warning message to the console. The purpose of this option is to provide notification of a long-running UR. Long-running URs may result in a lengthy DB2 restart or a lengthy recovery situation for critical tables. Specify the value in 1K (1000 log records) increments. A value of 0 indicates that there will be no write check.

## 10. LIMIT BACKOUT

Acceptable values:	AUTO, YES, NO
Default:	AUTO
Update:	option 23 on panel DSNTIPB
DSNZPxxx:	DSN6SYSP LBACKOUT

Specify whether to postpone some backward log processing.

*NO* specifies that DB2 backward log processing should process all inflight and inabort units of recovery (URs). *YES* postpones backout processing for some units of work until you issue the command RECOVER POSTPONED. *AUTO* postpones some backout processing, but automatically starts the backout processing when DB2 restarts and begins acceptance of new work. With YES or AUTO, backout processing runs concurrently with new work. Page sets or partitions with backout work pending are unavailable until their backout work is complete.

See the description of field BACKOUT DURATION for information on how you specify the amount of backward log processing that is allowed to complete during restart (when LIMIT BACKOUT is YES or AUTO).

The default value, AUTO, introduces a release incompatibility. Upon completing restart, some objects (table spaces, index spaces or partitions) might be unavailable until the automatically started process completes the backout work. In earlier releases, all of DB2 remains unavailable until backout work is complete. This incompatibility offers improved availability as the default.

## 11. BACKOUT DURATION

Acceptable values:	0-255
Default:	5
Update:	option 23 on panel DSNTIPB
DSNZPxxx:	DSN6SYSP BACKODUR

Specify a multiplier to indicate how much log to process for backout when LIMIT BACKOUT = YES or AUTO.

During restart, backward log processing continues until both of the following events occur:

- All inflight and inabort URs with update activity against the catalog or directory are backed out.
- The number of log records processed is equal to the number you specify in BACKOUT DURATION times the number of log records per checkpoint, which you specified in field CHECKPOINT FREQ. If the checkpoint frequency type is minutes, the default value of 50 000 log records will be used to calculate the number of log records to process.

Inflight and inabort URs that are not completely backed out during restart are converted to postponed abort status. Page sets or partitions with postponed backout work are put into restart pending (RESTP). This state blocks all access to the object other than access by the command RECOVER POSTPONED or by automatic backout processing performed by DB2 when LIMITED BACKOUT = AUTO.

A table space might be in restart pending mode, without the associated index spaces also in restart pending mode. This happens if a postponed abort UR makes updates only to non-indexed fields of a table in a table space. In this case, the indexes are accessible to Structured Query Language (SQL) (for index-only queries), even though the table space is inaccessible.

## 12. RO SWITCH CHKPTS

Acceptable values:	1-32767
Default:	5 (checkpoints)
Update:	option 23 on panel DSNTIPB
DSNZPxxx:	DSN6SYSP PCLOSEN

Indicates the number of consecutive DB2 checkpoints since a page set or partition was last updated after which DB2 converts the page set or partition from read-write to read-only. This value is used in conjunction with RO SWITCH TIME. If the condition for RO SWITCH TIME or RO SWITCH CHKPTS is met, the page set or partition is converted from read-write to read-only.

Having DB2 switch an infrequently-updated page set from read-write to read-only can be a performance benefit for recovery, logging, and for data sharing processing. See Part 5 (Volume 2) of *DB2 Administration Guide* and *DB2 Data Sharing: Planning and Administration* for more information about read-only switching.

### 13. RO SWITCH TIME

Acceptable values:	1-32767
Default:	10 (minutes)
Update:	option 23 on panel DSNTIPB
DSNZPxxx:	DSN6SYSP PCLOSET

Indicates the number of minutes since a page set or partition was last updated, after which DB2 converts the page set or partition from read-write to read-only. This value is used in conjunction with RO SWITCH CHKPTS. If the condition for RO SWITCH CHKPTS or RO SWITCH TIME is met, the page set or partition is converted from read-write to read-only.

Having DB2 switch an infrequently-updated page set from read-write to read-only can be a performance benefit for recovery, logging, and for data sharing processing. See Part 5 (Volume 2) of *DB2 Administration Guide* and *DB2 Data Sharing: Planning and Administration* for more information about read-only switching.

### 14. LEVELID UPDATE FREQ

Acceptable values:	0-32767
Default:	5 (checkpoints)
Update:	option 15 on panel DSNTIPB
DSNZPxxx:	DSN6SYSP DLDFREQ

Controls how often, in number of checkpoints, the level ID of a page set or partition is updated. A zero (0) disables downlevel detection.

Consider the following questions when you choose a value for the frequency of level ID updates:

- **How often do you use backup and restore methods outside of DB2's control** (such as DSN1COPY or DFDSS dump and restore)? If you rarely use such methods, there is no need to update the level ID frequently.
- **How many page sets are open for update at the same time?** If DB2 updates level IDs frequently, you have extra protection against downlevel page sets. However, if the level IDs for many page sets must be set at every checkpoint, you might experience a performance degradation.
- **How often does the subsystem take checkpoints?** If your DB2 subsystem takes frequent system checkpoints, you can set the level ID frequency to a higher number.

# 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

2

PRIMARY QUANTITY

===>

3

SECONDARY QTY.

===>

4

CATALOG DATA

===>

NO

5

DEVICE TYPE 1

===>

TAPE

6

DEVICE TYPE 2

===>

7

BLOCK SIZE

===>

28672

8

READ TAPE UNITS

===>

2

9

DEALLOC PERIOD

===>

0

10

RECORDING MAX

===>

1000

11

WRITE TO OPER

===>

YES

12

WTOR ROUTE CODE

===>

1,3,4

13

RETENTION PERIOD

===>

9999

14

QUIESCE PERIOD

===>

5

15

COMPACT DATA

===>

NO

PRESS:

ENTER to continue

RETURN to exit

HELP for more information

Blk, Trk, or Cyl

Primary space allocation

Secondary space allocation

YES or NO to catalog archive data sets

Unit name for COPY1 archive logs

Unit name for COPY2 archive logs

Rounded up to 4096 multiple

Maximum allocated read tape units

Time interval to deallocate tape units

Number of data sets recorded in BSDS

Issue WTOR before mount for archive

Routing codes for archive WTORS

Days to retain archive log data sets

Maximum quiesce interval (1-999)

YES or NO for data compaction

Figure 32. 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. DFSMS Direct Access Device Space Management (DASDM) limits the space allocation on a single volume to less than 64 000 tracks. Therefore, if the archive log data set size can be 64 000 tracks or greater, you need to specify a primary space quantity of less than 64 000 tracks, which will force the archive log data set to extend to a second volume. For more information about estimating the archive log data set size, see “Disk requirements for the archive log data sets” on page 35.

## 3. SECONDARY QTY.

Acceptable values:1-99999999

#  
#  
#  
#  
#  
#  
#

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 can extend to a maximum of 15 volumes. 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 and secondary space allocations (fields 2 and 3) large enough to contain all of the data coming from the active log data sets without extending beyond 15 volumes.

When archiving to DASD, DB2 will use the number of online storage volumes for the specified UNIT name to determine a count of candidate volumes, up to a maximum of 15 volumes. If the archives will be SMS managed, do not use a storage class with the Guaranteed Space attribute. SMS will attempt to allocate a primary extent on every candidate volume. This can result in allocation failures or unused space because the primary extent on each unused volume is not released when the archive data set is closed.

#### 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 44 as a guide.

Table 44. 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.

**Recommendation:** 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, the recommendation is 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 Part 4 (Volume 1) of *DB2 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 *OS/390 MVS System Codes*. The routing codes are also discussed in the description of the WTO macro in *OS/390 MVS Programming: Assembler Services Reference*.

### 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 data.

For information about the archive log data sets and how they can be managed using the change log inventory utility (DSNJU003), refer to Part 4 (Volume 1) of *DB2 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 Part 4 (Volume 1) of *DB2 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 ==>                26 ==>
3  ==>                15 ==>                27 ==>
4  ==>                16 ==>                28 ==>
5  ==>                17 ==>                29 ==>
6  ==>                18 ==>                30 ==>
7  ==>                19 ==>                31 ==>
8  ==>                20 ==>                32 ==>
9  ==>                21 ==>                33 ==>
10 ==>                22 ==>                34 ==>
11 ==>                23 ==>                35 ==>
12 ==>                24 ==>                36 ==>
13 ==>                25 ==>                37 ==>

PRESS:  ENTER to continue  RETURN to exit  HELP for more information
```

Figure 33. 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 Part 4 (Volume 1) of *DB2 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 support for a communications network” on page 260 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      Status of a qualifying database access
                                           thread after commit. ACTIVE or INACTIVE.
 8 MAX TYPE 1 INACTIVE   ====> 0          Max number of type 1 inactive threads.
 9 DB2 GENERIC LUNAME    ====>             Generic VTAM LU name for this DB2
                                           subsystem or data sharing group.
10 IDLE THREAD TIMEOUT  ====> 0          0 or seconds until dormant server ACTIVE
                                           thread will be terminated (0-9999)
11 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 34. 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 sample jobs (DSNTEJ6S, DSNTEJ6P, DSNTEJ6, DSNTEJ6D, DSNTEJ6T, DSNTEJ63, DSNTEJ64, and DSNTEJ65) 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). The BSDS is updated by the change log inventory in step DSNTLOG of install job DSNTIJUZ.

## 2. DB2 LOCATION NAME

Acceptable values:	1-16 alphanumeric characters
Default:	LOC1
Update:	see Update on page 233
DSNZPxxx:	none

Specify the unique name which 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 233
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 233
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 Part 5 (Volume 2) of *DB2 Administration Guide*.

For more information on using the governor for remote queries, see Part 5 (Volume 2) of *DB2 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 cursors.

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.

If you specify INACTIVE, then DB2 supports two different types of inactive threads:

- A **type 1** inactive thread, which has the same characteristics as inactive threads available in releases prior to DB2 for OS/390 and z/OS Version 7.
- A **type 2** inactive thread, which uses less storage than a type 1 inactive thread.

Because they use less storage, type 2 inactive threads are more desirable than type 1 inactive threads. However, not all inactive threads can be of type 2 based on what type of operations the thread is performing, as summarized in Table 45. If a thread is to become inactive, DB2 tries to make it a type 2 inactive thread. If DB2 cannot make it a type 2 inactive thread, it tries to make it a type 1 inactive thread. If neither attempt is successful, then the thread remains active.

Table 45. Requirements for type 1 and type 2 inactive threads

If there is...	Thread can be type 2	Thread can be type 1
a hop to another location	Yes	Yes
a connection using DB2 private protocols	No	Yes
a package bound with RELEASE(COMMIT)	Yes	Yes
a package bound with RELEASE(DEALLOCATE)	Yes	No
a held cursor, a held LOB locator, or a package bound with KEEP_DYNAMIC(YES)	No	No

See Part 5 (Volume 2) of *DB2 Administration Guide* for more information about active and inactive threads.

## 8. MAX TYPE 1 INACTIVE

Acceptable values:	0 - value of MAX REMOTE CONNECTED
Default:	0
Update:	option 26 on panel DSNTIPB
DSNZPxxx:	DSN6FAC MAXTYPE1

The value in this field specifies the number of type 1 inactive threads that DB2 allows. This limit is defined because a large number of type 1 inactive threads might adversely affect system performance. Type 1 inactive threads are used for private protocol. DRDA uses type 2 inactive threads.

A value of zero indicates that type 1 inactive connections are not allowed. If a thread meets the requirement for being made a type 1 inactive thread and MAX TYPE 1 INACTIVE is zero, the thread remains active.

A value of greater than zero indicates that type 1 inactive connections are allowed, but are limited to the specified number. When a thread meets the requirement for being made a type 1 inactive thread, and MAX TYPE 1 INACTIVE is reached, the remote connection is terminated.

If you want type 1 inactive connections to be allowed, set this value to the maximum number of concurrent connections that you want to allow to go inactive that access another remote location with three part names.

A value equal to the value in field MAX REMOTE CONNECTED from panel DSNTIPE allows all remote threads to become type 1 inactive threads.

## 9. 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 13. Connecting distributed database systems" on page 465 for more information about setting up a generic LU name.

## 10. IDLE THREAD TIMEOUT

Acceptable values:	0-9999
Default:	0
Update:	option 26 on panel DSNTIPB
DSNZPxxx:	DSN6FAC IDTHTOIN

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 7. (Idle server threads remain in the system and continue to hold their resources, if any.)

## 11. 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 DRDArequesters 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          INSTALL DB2 - 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
 4 EXTRA BLOCKS REQ   ===> 100      Maximum extra query blocks when DB2 acts
                                     as a requester. 0-100
 5 EXTRA BLOCKS SRV   ===> 100      Maximum extra query blocks when DB2 acts
                                     as a server. 0-100
 6 DATABASE PROTOCOL  ===> DRDA     Protocol used for three-part name if
                                     not specified on BIND. DRDA or PRIVATE.
 7 AUTH AT HOP SITE    ===> BOTH    Authorization at hop site. BOTH or RUNNER.
 8 TCP/IP KEEPALIVE    ===> ENABLE   ENABLE, DISABLE, or 1-65534
 9 POOL THREAD TIMEOUT ===> 120     0-9999 seconds

PRESS: ENTER to continue  RETURN to exit  HELP for more information
```

Figure 35. 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 *DB2 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 *DB2 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 Kerberos 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 Part 3 (Volume 1) of *DB2 Administration Guide* for more information.

### 4. EXTRA BLOCKS REQ

Acceptable values:	0-100
Default:	100
Update:	option 27 on panel DSNTIPB
DSNZPxxx:	DSN6SYSP EXTRAREQ

The value in this field establishes an upper limit on the number of extra DRDA query blocks DB2 requests from a remote DRDA server. This does not limit the size of the SQL query answer set. It simply controls the total amount of data that can be transmitted on any given network exchange.

### 5. EXTRA BLOCKS SRV

Acceptable values:	0-100
Default:	100
Update:	option 27 on panel DSNTIPB
DSNZPxxx:	DSN6SYSP EXTRASRV

The value in this field establishes an upper limit on the number of extra DRDA query blocks DB2 returns to a DRDA client. This does not limit the size of the SQL query answer set. It simply controls the total amount of data that can be transmitted on any given network exchange.

### 6. DATABASE PROTOCOL

Acceptable values:	DRDA or PRIVATE
Default:	DRDA
Update:	option 27 on panel DSNTIPB
DSNZPxxx:	DSN6SYSP DBPROTCL

The value in this field determines the default protocol (DRDA or PRIVATE) that is to be used when option DBPROTOCOL BIND is not explicitly specified for the bind of a plan or a package. If you selected INSTALL for field INSTALL TYPE on panel DSNTIPA1, the default value for DATABASE PROTOCOL is DRDA. If you selected MIGRATE for field INSTALL TYPE on panel DSNTIPA1, the default value for DATABASE PROTOCOL is PRIVATE.

An application program might contain statements with three-part names, or aliases that reference remote objects. At bind or rebind of a plan, a user can specify whether these statements flow to the remote site using DB2 private protocol or DRDA protocol when communicating with the remote server.

Choose a value of PRIVATE under either of the following conditions:

- Your DB2 Version 7 subsystem is in a data sharing group with DB2 subsystems that are at previous release levels, and you plan to bind plans or packages on the Version 7 subsystem. If you choose a default value of DRDA for your Version 7 data sharing member, and you use the BIND subcommand to bind plans or packages on the Version 7 member, those plans and packages are implicitly bound with the bind option DBPROTOCOL(DRDA), even though you did not specify that option. Those plans and packages become frozen objects at the DB2 Version 7 level and cannot be run on the DB2 members that are at previous levels. This is true whether or not the application uses three-part names.
- You do not plan to move applications that use three-part names to DRDA access immediately. To use DRDA access for applications with three-part names, you must bind packages for those applications at each location that the applications access, then bind all packages into a plan. If you cannot perform this activity immediately, and you want your applications to continue to work, you should specify PRIVATE for DATABASE PROTOCOL. See chapter 5 of *DB2 Application Programming and SQL Guide* for more information on moving applications from DB2 private protocol access to DRDA access.

If you choose a value of DRDA, and your DB2 Version 7 subsystem is in a data sharing group with DB2 subsystems that are at previous release levels, you need to bind the plans for DB2-supplied applications such as SPUFI and DCLGEN with the DBPROTOCOL(PRIVATE) option so that those applications are accessible to other members of the data sharing group. The BIND commands for DB2-supplied applications are in job DSNTIJSG. For information on job DSNTIJSG, see “Migration step 21: Bind SPUFI and DCLGEN and user maintained database activity: DSNTIJSG” on page 301 or “Migration step 21: Bind SPUFI and DCLGEN and user maintained database activity: DSNTIJSG” on page 339.

## 7. AUTH AT HOP SITE

Acceptable values:	BOTH or RUNNER
Default:	BOTH
Update:	option 27 on panel DSNTIPB
DSNZPxxx:	DSN6SPRM HOPAUTH

The value in this field indicates whose authorization is checked at a second server (sometimes called a “hop” site) when the request is from a requester that is not DB2 for OS/390 and z/OS. This option applies only when private protocol is used for the hop from the second to third site. See topic Privileges exercised through a plan or package in Part 3 of *DB2 Administration Guide* for more information about authorization for distributed processing.

BOTH, the default, means that the package owner’s authorization is checked for static SQL, and the runner’s authorization ID is checked for dynamic SQL. (In Version 5, this value was YES.)

RUNNER means that both static and dynamic SQL use the runner’s authorization. (In Version 5, this value was NO.)

## 8. TCP/IP KEEPALIVE

Acceptable values:	ENABLE, DISABLE, or 1-65534
Default:	ENABLE
Update:	option 27 on panel DSNTIPB
DSNZPxxx:	DSN6FAC TCPKPALV

In cases where the TCP/IP KeepAlive value in the TCP/IP configuration is not appropriate for the DB2 subsystem, this field can be used as an override. The following values can be specified:

- **ENABLE**

Do not override the TCP/IP KeepAlive configuration value. This is the default.

- **DISABLE**

Disable KeepAlive probing for this subsystem.

- **1-65534**

Override the TCP/IP KeepAlive configuration value with the number entered. This value is specified in seconds. Consider setting this value close to the IDLE THREAD TIMEOUT value on installation panel DSNTIPR or the IRLM RESOURCE TIMEOUT value on installation panel DSNTIPI.

Avoid using very small values. KeepAlive detection is accomplished by probing the network based on the time entered in the KeepAlive parameter. A small KeepAlive value can cause excessive network traffic and system resource consumption. Maintain a proper balance that allows network failures to be detected on a timely basis yet does not severely impact system and network performance. See Part 5 (Volume 2) of *DB2 Administration Guide* for more information.

## **9. POOL THREAD TIMEOUT**

Acceptable values:	0-9999
Default:	120
Update:	option 27 on panel DSNTIPB
DSNZPxxx:	DSN6FAC POOLINAC

Specify the approximate time, in seconds that a data base access thread (DBAT) can remain idle in the pool before it is terminated. A data base access thread in the pool counts as an active thread against MAX REMOTE ACTIVE and may hold locks, but does not have any cursors. Because threads are checked every 3 minutes to see if they have remained in the pool past the timeout limit, timeout values of less than 3 minutes may not be honored for up to 3 minutes.

Specifying 0 causes a DBAT to terminate rather than go into the pool if there is a sufficient number of threads in the pool to process the number of type 2 inactive threads that currently exist.

---

## Routine parameters panel: DSNTIPX

The entries on this panel are used to start the stored procedures address space to run stored procedures or user-defined functions. See “Enabling stored procedures after installation” on page 266 if you want to enable stored procedures outside of the standard installation steps.

DSNTIPX

INSTALL DB2 - ROUTINE PARAMETERS

===>\_

Enter data below:

1 WLM PROC NAME

2 DB2 PROC NAME

3 NUMBER OF TCBS

4 MAX ABEND COUNT

5 TIMEOUT VALUE

6 WLM ENVIRONMENT

====> DSN1WLM

====> DSN1SPAS

====> 8

====> 0

====> 180

====>

WLM-established stored procedure JCL PROC

DB2-established stored procedure JCL PROC

Number of concurrent TCBS (1-100)

Allowable ABENDs for a routine (0-255)

Seconds to wait before SQL CALL or

function invocation fails (5-1800,NOLIMIT)

Default WLM environment name

PRESS:

ENTER to continue

RETURN to exit

HELP for more information

Figure 36. Routine 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, DSNTEJ6S, DSNTEJ6P, DSNTEJ6T, DSNTEJ6D, DSNTEJ63, DSNTEJ64, and DSNTEJ65.

### 3. NUMBER OF TCBS

Acceptable values:	1-100
Default:	8



DSNZPxxx: none

Specify how many SQL CALL statements or user-defined invocations can be processed concurrently in one address space. The larger the value, the more stored procedures and user-defined functions you can run concurrently in one address space. For information about how this value affects storage below the 16MB line, see Part 5 (Volume 2) of *DB2 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 or an invocation of a user-defined function is allowed to terminate abnormally, after which SQL CALL statements for the stored procedure or user defined function are rejected. The default, 0, means that the first abend of a stored procedure or user defined function causes SQL CALLs to that procedure or function 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 or invocation of a user-defined function 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 or the user defined function does not complete, your SQL request is hung until the request is satisfied or the thread is canceled.

#### 6. WLM ENVIRONMENT

Acceptable values: Any valid name from 1-18 alphanumeric characters, including underscores  
Default:  
Update: option 28 on panel DSNTIPB  
DSNZPxxx: DSN6SYSP WLMENV

The value in this field specifies the name of the WLM\_ENVIRONMENT to use for a user-defined function or stored procedure when a value is not given for the WLM\_ENVIRONMENT option on the CREATE FUNCTION or CREATE PROCEDURE statements. The name can include the underscore character.

Changing this value does not change existing routines because the value is stored in the catalog when the function or procedure is created.

## 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 Part 3 (Volume 1) of *DB2 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 37. 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 Part 3 (Volume 1) of *DB2 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 RGFDDBNAM

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 RGFMNORT

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 38. Job editing panel: DSNTIPY

### 1. REMOTE LOCATION

Acceptable values:	1-16 characters
Default:	none
Update:	See “The update process” on page 231
DSNZPxxx:	none

Specify the location name of another DB2 for OS/390 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, DSNTEJ6D, DSNTEJ63, DSNTEJ64, and DSNTEJ65). 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 231
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 355 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 231
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.

An example of job card information is below:

```
3====> // MEMBER JOB,  
4====> // MSGLEVEL=(1,1),MSGCLASS=H,REGION=4096,CLASS=A  
5====> // USER=SYSADM,PASSWORD=SYSADM,NOTIFY=SYSADM  
6====>
```

---

## 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 Part 2 of *DB2 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 = irlmreg K, AVAILABLE = irlmav 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:

$$70 \times 10000 \times 250 = 1750000\text{KB}$$

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.



```

DSNTIPC          INSTALL DB2 - CLIST CALCULATIONS - PANEL 1
====>

You can update the DSMAX, EDMPOOL, EDMPOOL DATA SPACE SIZE/MAX
(if CACHE DYNAMIC is YES), SORT POOL, and RID POOL sizes if necessary.

                                     Calculated  Override

1  DSNTPC - MAXIMUM OPEN DATA SETS   =    3000          (1-32767)

2  DSNTPC5I EDMPOOL STORAGE SIZE      =   14812 K          K
3  DSNTPC5I EDMPOOL DATA SPACE SIZE  =     0 K          K
4  DSNTPC5I EDMPOOL DATA SPACE MAX   =     0 K          K
5  DSNTPC5I BUFFER POOL STORAGE SIZE  =    8768 K          K
6  DSNTPC5I SORT POOL SIZE            =    1000 K          K
7  DSNTPC5I RID POOL SIZE             =    4000 K          K
8  DSNTPC5I DATA SET STORAGE SIZE    =    5400 K          K
9  DSNTPC5I CODE STORAGE SIZE         =    4300 K          K
10 DSNTPC5I WORKING STORAGE SIZE       =    5960 K          K
11 DSNTPC6I TOTAL MAIN STORAGE        =   44240 K          K
12 DSNTPC4I RECOMMENDED REAL STORAGE  =   42164 K          K
13 DSNTPC7I TOTAL STORAGE BELOW 16M  =    1634 K (WITH SWA ABOVE 16M LINE)
14 DSNTPC3I IRLM LOCK MAXIMUM SPACE   =  335000 K, AVAILABLE = 6144 + K

PRESS:  ENTER to continue RETURN to exit  HELP for more information

```

Figure 39. CLIST calculations panel 1: DSNTIPC

## 1. DSMAX

Acceptable values: 1-32767  
Default: based on calculations  
Update: enter a value in the override column  
DSNZPxxx: DSN6SPRM DSNMAX

This field specifies the maximum number of data sets that can be open at one time. The practical limit may be less than 32727, depending on available storage below the line. The value you enter can substantially influence the performance of DB2; see Part 5 (Volume 2) of *DB2 Administration Guide* for more information.

## 2. EDMPOOL STORAGE SIZE

Acceptable values: 1-2097152  
Default: based on calculations  
Update: press ENTER twice on panel DSNTIPB  
DSNZPxxx: DSN6SPRM EDMPOOL

This field specifies the size of the environmental descriptor manager (EDM) pool calculated by the CLIST in kilobytes. 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.

For information on how DB2 calculates the EDM pool size, see “EDM pool size calculation” on page 43.

## 3. EDMPOOL DATA SPACE SIZE

Acceptable values: 1-2097152

Default:	based on calculations
Update:	press ENTER on panel DSNTIPB
DSNZPxxx:	DSN6SPRM EDMDSPAC

This field specifies the size (in kilobytes) of the data space that can be used by the environmental descriptor manager (EDM) pool. The CLIST calculates a data space size only if you specified YES for CACHE DYNAMIC SQL on panel DSNTIP4, otherwise it is zero. 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, unless CACHE DYNAMIC SQL is NO, in which case any value in this column is ignored and is blanked out by the install CLIST.

For information on how DB2 calculates the EDM data space pool size, see “EDM pool size calculation” on page 43.

#### 4. EDMPOOL DATA SPACE MAX

Acceptable values:	0-2097152
Default:	1048576
Update:	press ENTER on panel DSNTIPB
DSNZPxxx:	DSN6SPRM EDMDSMAX

This field specifies the MAXIMUM size (in kilobytes) of the data space that can be used by the environmental descriptor manager (EDM) pool. The CLIST calculates a maximum data space size only if you specified YES for CACHE DYNAMIC SQL on panel DSNTIP4. If you did not specify YES, the maximum data space size is zero.

Then choose one of the following:

- Accept the value in the calculated column. If there is a value in the override column, you must erase the override value in order to accept the calculated value.
- Type your own value in the override column. If CACHE DYNAMIC SQL is NO, any value in this column is ignored and blanked out by the install CLIST.

#### 5. BUFFER POOL STORAGE 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. This field is protected and cannot be changed during update mode. If you want to change the size of a buffer pool, you must use command -ALTER BUFFERPOOL as described in Chapter 2 of *DB2 Command Reference*.

#### 6. SORT POOL SIZE

Acceptable values:	240K-64000K
Default:	1MB
Update:	press ENTER twice on panel DSNTIPB

DSNZPxxx: DSN6SPRM SRTPOOL

This field specifies the amount of storage needed for the sort pool. You have a choice of:

- Accepting the default value in the *Calculated* column. If there is a value in the *Override* column, you must erase the override value in order to accept the default 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:

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

See Part 5 (Volume 2) of *DB2 Administration Guide* for detailed instructions on choosing sizes for optimal performance.

## 7. RID POOL SIZE

Acceptable values: 0, 16K-1000000K  
Default: 4MB  
Update: press ENTER twice on panel DSN TIPB  
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 default value in the *Calculated* column. If there is a value in the *Override* column, you must erase the override value in order to accept the default 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:

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

where *A* is:

- 4 for non-large tables
- 5 for large tables

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 Part 5 (Volume 2) of *DB2 Administration Guide* for how to choose a RID pool size for optimal performance.

## 8-12. CLIST messages

Acceptable values: none  
Default: none  
Update: run CLIST again  
DSNZPxxx: none

I                    These fields specify sizes calculated by the CLIST. Fields 8 through 12 are protected and cannot be changed.

I                    **13-14. 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 volser1 WILL REQUIRE AT LEAST 1000 4K BLOCKS
2 DSNT488I  VOLUME volser2 WILL REQUIRE AT LEAST 1000 4K BLOCKS
3 DSNT488I  VOLUME volser3 WILL REQUIRE AT LEAST 5970 4K BLOCKS
4 DSNT488I  VOLUME volser4 WILL REQUIRE AT LEAST 46715 4K BLOCKS
5 DSNT488I  VOLUME volser5 WILL REQUIRE AT LEAST 15758 4K BLOCKS
6 DSNT488I  VOLUME volser6 WILL REQUIRE AT LEAST 25995 4K BLOCKS
7 DSNT488I  VOLUME volser7 WILL REQUIRE AT LEAST 25995 4K BLOCKS

PRESS:  ENTER to select  RETURN to exit  HELP for more information
```

Figure 40. 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 Part 3 of *DB2 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 222. (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	DSNTIJTM	DSNTIJSG	DSNTIJIC
DSNTIJDE	DSNTEJ0	DSNTEJ1	DSNTEJ1L	DSNTEJ1P
DSNTEJ1S	DSNTEJ1T	DSNTEJ2A	DSNTEJ2C	DSNTEJ2D
DSNTEJ2E	DSNTEJ2F	DSNTEJ2P	DSNTEJ2U	DSNTEJ3C
DSNTEJ3P	DSNTEJ4C	DSNTEJ4P	DSNTEJ7	DSNTEJ71

DSNTEJ73	DSNTEJ75	DSNTESA	DSNTESC	DSNTESD
DSNTESE	DSNTEJ1U	DSNTEJ61	DSNTEJ62	DSNTEJ6V

If you have activated DDF, the CLIST also tailors job DSNTEJ6.

If you have activated stored procedures, DSNTEJ6D, DSNTEJ6S, DSNTEJ6P, DSNTEJ6T, DSNTEJ6U, DSNTEJ6W, DSNTEJ6Z, DSNTEJ63, DSNTEJ64, and DSNTEJ65 are edited.

If CICS is selected, the CLIST edits DSNTEJ5A, DSNTEJ5C, and DSNTEJ5P.

If you are using data sharing, DSNTIJGF and DSNTIJFT are also edited.

The installation CLIST tailors the DSNHC, 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
DSNTEJ7	DSNTEJ71	DSNTEJ73	DSNTEJ75	DSNTEJ2U
DSNTESA	DSNTESC	DSNTESD	DSNTESE	DSNTIJEX
DSNTIJFV	DSNTIJIC	DSNTIJIN	DSNTIJMV	DSNTIJSG
DSNTIJTC	DSNTIJTM	DSNTIJUZ	DSNTIJVC	DSNTEJ1L
DSNTEJ2U	DSNTEJ61	DSNTEJ62	DSNTEJ6V	DSNTEJ6W
DSNTEJ6Z				

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, DSNTEJ6T, DSNTEJ6U, DSNTEJ63, DSNTEJ64, and DSNTEJ65 are edited.

If CICS is selected, 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 *DB2 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 IEBPTPCH 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 222.

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 *DB2 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, the recommendation is to have a temporary library, or one per user, rather than trying to share libraries. However, if your DB2 subsystem uses DFSMSdfp'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, the recommendation is to have a temporary library, or one per user, rather than trying to share libraries.
- PLIB(NONE)



- Check default processor options. If you prefer other default options, change them. The following are the default processor options:

CICSOPT(NONE)	LOPTION(NONE)
COPTION(NONE)	PASS(DEFAULT)

- Check print and work space defaults. If the default allocation sizes are not acceptable for your site, change them. The following are the print and work space defaults:

PSECSPAC(20)	WSECSPAC(20)
PSPACE(20)	WSPACE(20)
WORKUNIT(DEFAULT)	

4. Examine the data set names for other products. Many data set names for other products appear in the jobs. These names are shown in Table 35 on page 112. Change them if they are different at your site.

## Editing the subsystem parameters and DSNHDECP values

The subsystem parameter module is generated by job DSNTIJUZ each time you install, migrate, or update DB2. Seven macros expand to form this 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. For more information, see “Installation step 5: Define DB2 initialization parameters: DSNTIJUZ” on page 242 or “Migration step 11: Define DB2 initialization parameters: DSNTIJUZ” on page 293.

The data-only load module DSNHDECP is also generated by job DSNTIJUZ. It contains the application programming defaults.

For a directory of subsystem parameters and DSNHDECP values, see “Appendix B. Directory of subsystem parameters” on page 533. Version 7 allows online changes to many of the subsystem parameters. The new SET SYSPARM command will enable a function that allows reloading. For more information, see *DB2 Command Reference*.

---

## 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 107 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 SIZES *                        24 ARCHIVE LOG DATA SET PARAMETERS
10 SIZES PANEL 2                  25 DATABASES TO START AUTOMATICALLY
11 THREAD MANAGEMENT              26 DISTRIBUTED DATA FACILITY PANEL
12 BUFFER POOL SIZES PANEL 1      27 DISTRIBUTED DATA FACILITY PANEL 2
13 BUFFER POOL SIZES PANEL 2 *    28 ROUTINE PARAMETERS
14 BUFFER POOL SIZES PANEL 3 *    29 DATA DEFINITION CONTROL SUPPORT
15 TRACING AND CHECKPOINT PARAMETERS 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 41. Individual update menu panel: DSNTIPB

On the command line, enter a number to select the family of parameters you wish to update. These numbers correspond to the installation panels in Table 46.

Table 46. Panel identifiers

Panel ID	Panel Title	Page
1. DSNTIPA2	Data Parameters	96
2. DSNTIPK	Define Group or Member	101
3. DSNTIPH	System Resource Data Set Names	103
4. DSNTIPT	Data Set Names Panel 1	107
5. DSNTIPU	Data Set Names Panel 2	112
6. DSNTIPQ	Data Set Names Panel 3	120
7. DSNTIPG	Data Set Names Panel 4	123
8. DSNTIPW	Data Set Names Panel 5	127
9. DSNTIPD	Sizes	130

Table 46. Panel identifiers (continued)

Panel ID	Panel Title	Page
10. DSNTIP7	Sizes Panel 2	136
11. DSNTIPE	Thread Management	138
12. DSNTIP1	Buffer Pool Sizes Panel 1	143
13. DSNTIP2	Buffer Pool Sizes Panel 2	145
14. DSNTIP6	Buffer Pool Sizes Panel 3	147
15. DSNTIPN	Tracing Parameters	149
16. DSNTIPO	Operator Functions	153
17. DSNTIPF	Application Programming Defaults Panel 1	160
18. DSNTIP4	Application Programming Defaults Panel 2	168
19. DSNTIPI	IRLM Panel 1	174
20. DSNTIPJ	IRLM Panel 2	179
21. DSNTIPP	Protection	184
22. DSNTIPM	MVS PARMLIB Updates	188
23. DSNTIPL	Active Log Data Set Parameters	192
24. DSNTIPA	Archive Log Data Set Parameters	197
25. DSNTIPS	Databases and Spaces to Start Automatically	203
26. DSNTIPR	Distributed Data Facility	205
27. DSNTIP5	Distributed Data Facility Panel 2	210
28. DSNTIPX	Stored Procedures Parameters	214
29. DSNTIPZ	Data Definition Control Support	216
30. DSNTIPY	Job Editing	219

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 Part 2 (Volume 1) of *DB2 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 480.
  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 Part 4 (Volume 1) of *DB2 Administration Guide* for more information.
- To change the number of data sets for active logs, you can use the DSNJU003 utility. See Part 4 (Volume 1) of *DB2 Administration Guide* for details on the system programmer action in recovery scenarios for active log failures.

---

## Chapter 7. 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 51. You must also run the installation CLIST. It is described beginning on page 77.

If you plan to have data sharing enabled (Enable option on panel DSNTIPP1), you should save all your jobs from the original installation/migration in a different data set than the enable jobs. If you save them in *prefix.NEW.SDSNSAMP*, the jobs may be overwritten. For more information, see *DB2 Data Sharing: Planning and Administration*.

Before proceeding with the installation steps, refer to *IBM Database 2 Program Directory* shipped with the product for keyword specifications used 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 *OS/390 MVS Initialization and Tuning Reference*. Use the following parameters for DSNYASCP, DXRRLM00, and DSNUTILB:

Table 47. 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 SYS1.LINKLIB, 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.

**Recommendation:** 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 should be the next entry.

The DB2 entry has the following format:

```
SUBSYS  SUBNAME(ssname)
        INITRTRN(DSN3INI)
        INITPARM
        ('DSN3EPX,prefix<,scope<,group-attach>>')
```

where:

*ssname*            The DB2 subsystem name.

<b>DSN3INI</b>	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).
<b>DSN3EPX</b>	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).
<i>prefix</i>	<p>The 1- to 8-character command prefix.</p> <p>The first character of the command prefix must be one of the following: @ # . / ' ) * + - = ¢ &lt;   &amp; ! ; % _ ? : ". The remaining characters of the command prefix must be one of the above characters, A-Z, or 0-9. See Table 43 on page 189 more information. Do not use the JES2 backspace character or command prefix character. The default is the hyphen (-).</p> <p>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.</p>
<i>scope</i>	<p>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:</p> <p><b>S</b> Started, and register the prefix with Sysplex scope at DB2 startup instead of during MVS IPL. This is the default.</p> <p><b>Recommendation:</b> 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.</p> <p><b>M</b> MVS system scope, and register the prefix during MVS IPL.</p> <p><b>X</b> 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.</p> <p>For more information about the command prefix facility of MVS, see <i>OS/390 MVS Planning: Operations</i>.</p>
<i>group-attach</i>	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 LNKLISTxx. 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.

- LNKLISTxx

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 57.

Job DSNTIJMV updates this member to include the DB2 load module library, *prefix.SDSNLINK*, in the LNKLISTxx. If you moved the modules from *prefix.SDSNLINK* into another library, edit DSNTIJMV to include that library in the LNKLISTxx. If you have combined *prefix.SDSNLINK* and *prefix.SDSNLOAD* into one library, edit DSNTIJMV to include the combined library in the LNKLISTxx. See *OS/390 MVS Initialization and Tuning Guide* for restrictions on data sets that are concatenated in LNKLIST.

Any data set that is added to the LNKLISTxx 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 LNKLISTxx member.

If you do not include the DFSORT library in the LNKLISTxx 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 LNKLISTxx) 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 Part 5 of *DB2 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).
- MVS Workload Manager (WLM) procedure

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, DSNHASM, DSNHSQL) 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 48.

---

## 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 as explained on page 241.

---

## 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 disk space for your system. See Part 5 (Volume 2) of *DB2 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.

**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 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 untailored version of *prefix.SDSNSAMP*.
- Initializes the DB2 catalog database with data from the untailored *prefix.SDSNSAMP*.

- Invokes the DSNJLOGF utility to preformat active log data sets that are created during installation. See *DB2 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, remove the catalog delete statements from job DSNTIJDE, 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, job DSNTIJUZ does the following:

- Link-edits the assembled modules into the *prefix*.SDSNEXIT library.
- 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 and the DSNHDECM macro to move your subsystem name and the application programming 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.

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.

The subsystem parameter, PTASKROL in macro DSN6SYSP, indicates whether to roll up query parallel task's accounting trace records into the originating task's accounting trace. A value of YES means the originating task will generate an additional accounting trace record with all the roll-up values from parallel tasks.

**Recommendation:** Use the default value of YES. A value of NO means each parallel task will produce its own accounting trace record.

# The subsystem parameter OJPERFEH, in macro DSN6SPRM, lets you specify  
# whether to disable performance enhancements for outer join operations. See Part 5  
# (Volume 2) of *DB2 Administration Guide* for more information about how to set  
# OJPERFEH.

The subsystem parameters, SMSCDFL and SMSCDIX, in macro DSN6SPRM, specify whether SMS data class names are used for table spaces and indexes. SMSCDFL and SMSCDIX are one to eight characters long with a default value of a blank. When you use DFSMSHsm and DB2 storage groups, you can use the subsystem parameters SMSDCFL and SMSDCIX to assign table spaces and indexes to different DFSMSHsm data classes.

- SMSDCFL specifies a DFSMSHsm data class for table spaces. If you assign a value to SMSDCFL, DB2 specifies that value when it uses Access Method Services to define a data set for a table space. If the value of SMSDCFL is one or more blanks DB2 does not specify a data class when it creates data sets for table spaces.
- SMSDCIX specifies a DFSMSHsm data class for indexes. If you assign a value to SMSDCIX, DB2 specifies that value when it uses Access Method Services to define a data set for an index. If the value for SMSDCIX is one of more blanks, DB2 does not specify a data class when it creates data sets for indexes.

The default value for both subsystem parameters is a blank.

Before you set the data class system parameters, you need to define the data classes for your table space data sets and index data sets. You also need to code SMS ACS routines to assign indexes to one SMS storage class and table spaces to a different SMS storage class. For more information on creating data classes, see *DFSMS/MVS Storage Management Library: Implementing System-Managed Storage*.

The subsystem parameter NPGTHRSH, in macro DSN6SPRM, lets you specify that DB2 uses special access path selection for tables under a given size. The default value is 0 which means no special access path selection is used. See Part 5 (Volume 2) of *DB2 Administration Guide* for more information on how to set NPGTHRSH.

| The subsystem parameter STARJOIN, in macro DSN6SPRM, lets you specify when  
| DB2 enables star join processing. See Part 5 (Volume 2) of *DB2 Administration*  
| *Guide* for more information on how to set STARJOIN.

# The subsystem parameter OPTOPSE, in macro DSN6SPRM, lets you specify  
# whether DB2 can perform certain sort operations in parallel for queries that involve  
# join operations. See Part 5 (Volume 2) of *DB2 Administration Guide* for more  
# information on how to set OPTOPSE.

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 HDB7710, 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 you run steps DSNTLOG and DSNTIMQ, or you might receive a return code of 4 when you run 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 exit routines: DSNTIJEX (optional)

| Job DSNTIJEX builds the sample user exits, DSN3@SGN and DSN3@ATH, and  
| the user version of the access control user exit, DSNX@XAC, from the source code  
# in *prefix.SDSNSAMP* and places them in the *prefix.SDSNEXIT* library. DSNTIJEX  
# also assembles and link-edits the sample user exit DSNACICX. The DB2 CLIST  
# tailors the JCL in DSNTIJEX to match your site's environment.

The sample user exits are not the same as the default user exits supplied by DB2. By implementing the sample user exits you can provide group names as secondary authorization IDs. By modifying the sample user exits, you can tailor authorization processing for your subsystem.

| DSNXSXAC is a copy of the default access control user exit that can be modified  
| by the user. This exit allows you to bypass some or most of DB2 authorization  
| checking to specify your own authorization checking. If you do not modify it, this  
| step is not needed and should be deleted.

# DSNACICS is a stored procedure that invokes user exit DSNACICX, which you can  
# use to modify CICS parameters that the DSNACICS caller specifies. If you do not  
# need to modify the callerÆs parameter values, you can use the default DSNACICX  
# exit. However, if you need to modify the callerÆs parameter values, you need to  
# perform the following tasks:

- # 1. Write a user exit, in assembler, COBOL, C, or PL/I  
# 2. Assemble or compile the source code  
# 3. Link-edit the object code into the DB2 exit library

# Installation job DSNTIJEX includes a step to assemble and link-edit the sample  
# version of DSNACICX. You can use this step as a model for your program  
# preparation job.

For information on writing exit routines, see Appendix B (Volume 2) of *DB2 Administration Guide*. For more information on controlling data access, see Part 3 (Volume 1) of *DB2 Administration Guide*.

You have the following options regarding exit routines:

- To use the sample user exits, run job DSNTIJEX.
- To use the default user exits, skip job DSNTIJEX.
- To use the modified sample user exits, modify DSNTIJEX to reference the correct library before you run it.

If job DSNTIJEX runs successfully, it produces return codes of 4.

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.

- 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.
- 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 *DB2 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 *DB2 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 Part 5 (Volume 2) of *DB2 Administration Guide*.

For more information on SMF, refer to *OS/390 MVS Initialization and Tuning Reference* 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 Part 3 (Volume 1) of *DB2 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 7.

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 7 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 57 for information on link lists.

### Making DB2 CLISTs available to TSO and batch users: DSNTIJVC

From *prefix*.SDSNCLST, the 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 227 for information on the items you want to modify. The DSNEMC01 CLIST provides installation default values for the DB2I Default panels (option D on panel DSNEPRI). The first time a TSO user executes DB2I on a specific ISPF application such as DSNE (NEWAPPL), the DSNEMC01 CLIST sets the defaults based on the values specified on installation panel DSNTIPF. DSNEMC01 stores the values in the ISPF profile member DSNEPROF.



Job DSNTIJVC merges the tailored CLISTs from *prefix.NEW.SDSNTEMP* with unchanged CLISTs and REXX execs from *prefix.SDSNCLST*, and places all CLISTs and REXX execs 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 ISPPLIB, 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 48 shows the library table member names and their contents.

Table 48. The DB2 ISPF table library

DSNCONNS	ENABLE/DISABLE connection type and connection name variables that are referenced by plan or package name
DSNDBRMS	Subcommand DBRM member and LIBRARY name variables that are referenced by plan name
DSNPLPKN	Package list variables that are referenced by package name
DSNPATHS	Schema name variables that are referenced by plan or package name to be included in the PATH keyword

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
```

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 IKJ56861I.

For additional information on this TSO error message and how to *close* this data set, refer to *ISPF V4 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. If you chose IBMCOB for your LANGUAGE DEFAULT on panel DSNTIPF, you should also include the data set *prefix*.CEE.SCEERUN. For more information about the ISPF/CAF sample application, see “Running dynamic SQL and the ISPF/CAF application” on page 371.

## 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. **Recommendation:** Use 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 42 on page 250 and Figure 43 on page 252 display the needed modifications.

```

)ATTR
/*****
/* COPYRIGHT = 5740-XYR (C) COPYRIGHT IBM CORP 1982, 1985, 1990, 1999 */
/* REFER TO COPYRIGHT INSTRUCTIONS FORM NUMBER G120-2083 */
/* STATUS = VERSION 6, LEVEL 0 */
*****/
)BODY
%----- ISPF/PDF PRIMARY OPTION MENU -----
%OPTION ==>_ZCMD +USERID - &ZUSER .
% 0 +ISPF PARMs - Specify terminal and user parameters +TIME - &ZTIME .
% 1 +BROWSE - Display source data or output listings +TERMINAL - &ZTERM .
% 2 +EDIT - Create or change source data +PF KEYS - &ZKEYS .
% 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 +DB2I - Perform DATABASE 2 Interactive functions
% C +CHANGES - Display summary of changes for this release
% T +TUTORIAL - Display information about ISPF/PDF
% X +EXIT - Terminate ISPF using log and list defaults
%
+Enter%END+command to terminate ISPF.
%
)INIT
.HELP = ISR00003
&ZPRIM = YES /* ALWAYS A PRIMARY OPTION MENU */
&ZHTOP = ISR00003 /* TUTORIAL TABLE OF CONTENTS */
&ZHINDEX = ISR91000 /* TUTORIAL INDEX - 1ST PAGE */
)PROC
&ZSEL = TRANS( TRUNC (&ZCMD'.')
0,'PANEL(ISPOPTA)'
1,'PGM(ISRBRO)'
2,'PGM(ISREDIT)'
3,'PANEL(ISRUTIL)'
4,'PANEL(ISRFPA)'
5,'PGM(ISRJB1) PARM(ISRJPA) NOCHECK'
6,'PGM(ISRPTC)'
7,'PGM(ISRYXDR) NOCHECK'
8,'CMD(DSNECPRI) NEWAPPL(DSNE)'
C,'PGM(ISPTUTOR) PARM(ISR00005)'
T,'PGM(ISPTUTOR) PARM(ISR00000)'
' ',' '
X,'EXIT'
*, '?' )
&ZTRAIL = .TRAIL
)END

```

Figure 42. ISPF primary option panel (DSNTIPRM), edited to include DB2I

Panel DSNTIPRM is shown in Figure 42. Notice the added lines in boldface type. Adding these lines allows you to invoke the DB2 Interactive (DB2I) functions. The added lines include one displayed line:

```
% 8 +DB2I - Perform DATABASE 2 Interactive functions
```

and your choice of one of these undisplayed lines:

```

8,'CMD(DSNECPRI) NEWAPPL(DSNE)'
8,'CMD(DSNECPRI SSID(xxxx)) NEWAPPL(DSNE)'

```

The displayed line lets the user choose DB2I. Both of the undisplayed lines invoke the DB2I main panel (DSNEPRI). If you use the first undisplayed line, you accept the default for the subsystem identifier (SSID) parameter. If you use the second undisplayed lines, you can specify a different SSID parameter.

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. Any customized DSNEPROF members can be migrated from Version 6 to Version 7. **Recommendation:** Examine any new or changed default panel values to ensure that your custom values are still valid, specifically the option values for the subcommands BIND PLAN, REBIND PLAN, BIND PACKAGE, and REBIND PACKAGE.

**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-XYR (C) COPYRIGHT IBM CORP 1982, 1985, 1990, 1999 */
/* REFER TO COPYRIGHT INSTRUCTIONS FORM NUMBER G120-2083 */
/* STATUS = VERSION 6, 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 PARS    - 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 43. ISPF program development facility tutorial panel (DSNTIPTU), edited to include DB2 tutorial

The DSNTIPTU panel is shown in Figure 43. 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 V4 Dialog Developer's Guide and Reference* and *ISPF V4 User's Guide*.

## 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 11. Connecting the IMS attachment facility” on page 425. 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 CICS 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:

- 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 12. Connecting the CICS attachment facility” on page 433 along with the activities required to support CICS in a DB2 environment.

---

## Installation step 12: IPL MVS

The first time you start DB2, you must IPL MVS, then start the DB2 subsystem.

For Version 7, 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 235 and “Choosing link list options” on page 57.

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. See panel Figure 27 on page 174 for information about the MVS automatic restart manager.

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='ZPARM(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.

#  
#

VTAM must be defined before DDF can start. But, you do not need to have TCP/IP configured to start DDF. In addition, transactions such as those from DDF or CICS may fail because work files have not yet been defined.

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, and DSNTWR, which is the load module for the WLM\_REFRESH stored procedure. It also defines the initial buffer pool and hipool sizes as specified on installation panels DSNTIP1, DSNTIP2, and DSNTIP6.

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 219 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.DSNDBC.DSNDDB07.DSNkknn.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 work file 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 work file 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 “Disk requirements for the work file database” on page 29. 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 work file 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 49. DSNTIJTM return codes*

Step	PROCSTEP	Return code
DSNTIAD	PC	0000
	ASM	0000
	LKED	0000
DSNTWR	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.

I

You may receive a return code of 12 for step DSNTTMP if you have already run it.

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 including 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:

#  
#  
#

<b>DSNACICS</b>	CICS transaction invocation stored procedure that invokes CICS server programs
<b>DSNACCOR</b>	DB2 real-time statistics stored procedure
<b>DSNTPSMP</b>	SQL procedures processor stored procedure, which is used by the DB2 UDB Stored Procedure Builder

**DSNUTILS** Utility invocation stored procedure

**DSNWZP** Subsystem parameter settings stored procedure. For an example of how to invoke DSNWZP, see “Chapter 10. Verifying with the sample applications” on page 349.

**WLM\_REFRESH**

Stored procedure for refreshing a specified WLM environment

See Appendix B of *DB2 Utility Guide and Reference* for information on stored procedures that perform utility functions.

Stored procedures can be enabled after installation or migration. See “Enabling stored procedures after installation” on page 266 for the specific required steps.

The DSNTIJSJG job also does the following for user maintained database activity:

- Creates the resource limit facility database. See Part 5 (Volume 2) of *DB2 Administration Guide* for more information.
- Creates the data definition control support database. For more information, see Part 3 (Volume 1) of *DB2 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 210. 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 480 for VTAM information and “Step 4: Populate the communications database” on page 516 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 because 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 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 231 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)

DSNTIJSG 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:

- Part 5 (Volume 2) of *DB2 Administration Guide* for information on the resource limit specification table
- Part 3 (Volume 1) of *DB2 Administration Guide* for information on the data definition control support tables.

---

## Installation step 17: Bind the packages for DB2 REXX Language Support: DSNTIJRX (optional)

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.

**Note:** This job is not edited by the CLIST and will not be found in the SAMPLE LIBRARY from panel DSNTIPT (*prefix*.NEW.SDSNSAMP). DSNTIJRX is included in the target library if you have ordered DB2 REXX Language Support.

---

## 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.

**Recommendation:** 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 *DB2 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 257) to perform the authorization. For suggestions about using DB2 authorities and privileges, see Part 3 (Volume 1) of *DB2 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 10. Verifying with the sample applications” on page 349 .

---

## Installing support for a communications network

**Recommendation:** If you plan to use the distributed data facility (DDF), become familiar with the DDF function and install Virtual Telecommunications Access Method (VTAM) (required for DDF) and optionally OpenEdition® TCP/IP support before loading the data sets into DB2 libraries. For information on installing VTAM, see “Chapter 14. Connecting systems with VTAM” on page 469. For information on installing OpenEdition TCP/IP support, see “Chapter 15. Connecting systems with TCP/IP” on page 509.

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 Part 4 (Volume 1) of *DB2 Administration Guide* .

For information about customizing VTAM to work optimally with DB2, see “Step 1: Customize VTAM for DB2” on page 471.

**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 50 lists other considerations in planning for a second DB2 subsystem.

*Table 50. Considerations for installing a second DB2 subsystem*

Considerations	Decisions You Need to Make	Where to Find Information
Libraries	A single, shared library or separate libraries	Page 262
RACF protection	Resource and ID for the two subsystems	Note <sup>1</sup>
DB2 logging	BSDS, active, and archive log space requirements	Page 27
Database space requirements	DB2 directory, DB2 catalog, and user data	Note <sup>3</sup>
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 465
Common service area (CSA) requirement	DB2 and the IRLM	Page 36
Shared DASD	Giving different names to active and archive log data sets, or including those data sets in the GRS inclusion list	Page 262

Table 50. Considerations for installing a second DB2 subsystem (continued)

Considerations	Decisions You Need to Make	Where to Find Information
<b>Note:</b>		
<sup>1</sup> See Part 3 (Volume 1) of <i>DB2 Administration Guide</i>		
<sup>2</sup> See Part 5 (Volume 2) of <i>DB2 Administration Guide</i>		
<sup>3</sup> See the <i>DB2 Program Directory</i> .		

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 266 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 6. Installing, migrating, and updating system parameters” on page 77 to specify appropriate parameter values for the second subsystem. Table 51 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 51. 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.



Table 51. Parameters to change when installing the second subsystem (continued)

Installation Panel	Action
DSNTIP3	Check all values and make appropriate changes.
DSNTIPD	Check all values and make appropriate changes.
DSNTIP7	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.
DSNTIP6	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.
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
<b>DSNTIJCA</b>	Run this job if you are defining a new integrated catalog facility catalog.
<b>DSNTIJIN</b>	Run this job. It allocates the data sets listed below. These data sets contain the sample plans and programs. <i>prefix.DBRMLIB.DATA</i> <i>prefix.SRCLIB.DATA</i> <i>prefix.RUNLIB.LOAD</i> 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. LNKSTxx modifications are needed only for separate libraries.
- 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 Part 3 (Volume 1) of *DB2 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
<b>Logon procedure</b>	<p>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).</p> <p>A DB2 logon procedure can use only one set of DB2 libraries.</p>
<b>Concatenate CLISTs</b>	Update the SYSPROC library if you decide to have a separate library for the second subsystem.
<b>Concatenate panels</b>	Update the ISPPLIB library if you decide to have a separate library for the second subsystem.
<b>Concatenate messages</b>	Update the ISPMLIB library if you decide to have a separate library for a second subsystem.
<b>Customize panel</b>	You must update the ISPF primary option panel

(ISR@PRIM) if you decide to have a separate library for the second subsystem.

<b>Authorize users</b>	Grant authorization on both subsystems as necessary.
<b>DB2I defaults panel</b>	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>
<b>STEPLIB</b>	For separate libraries, change this DD statement to refer to the new libraries.
<b>DFSESL</b>	For separate libraries, change this DD statement to refer to the new libraries.
<b>Subsystem member</b>	Define a new subsystem name, language interface token (LIT), and command recognition character (CRC).
<b>Language interface</b>	Define a new LIT and reassemble.
<b>Linkage editor JCL</b>	Specify the library containing the new language interface.
<b>Authorize users</b>	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>
<b>IPL MVS</b>	This step is required to make any SYS1.PARMLIB changes take place.
<b>Start DB2</b>	Use the new command prefix (formerly called the subsystem recognition character) that you named during the installation process.
<b>DSNTIJIC</b>	Run this job.
<b>DSNTIJTM</b>	Run this job.
<b>DSNTIJSG</b>	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 DSNTPE71 on the first subsystem. <i>prefix.DBRMLIB.DATA</i> <i>prefix.RUNLIB.LOAD</i>
<b>DSNTEJxx</b>	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. <i>prefix.DBRMLIB.DATA</i> <i>prefix.SRCLIB.DATA</i> <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 10. Verifying with the sample applications” on page 349.

---

## 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.

---

## # Loading data with an SQL cursor

# You can use an SQL cursor with the LOAD utility to load data from a remote  
# location. Before loading data, bind the DSNUTIL package at each location from  
# which you want to load data. Customize and run job DSNTIJXL to bind the  
# packages. Sample job DSNTIJXL is located in data set DSN710.SDSNSAMP.

---

## 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 for the following:
    - 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 Language Environment 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 DSN710.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 Part 5 (Volume 2) of *DB2 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 *ssnmSPAS* PROC supplying the pertinent information: PROCNAME, SUBSYS name, NUMTCB, STEPLIB libraries. Use procedure DSNPAS in DSN710.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 DSN710.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 Part 5 (Volume 2) of *DB2 Administration Guide* for information on how to do this.
  - If the name of the WLM environment contains an underscore, the name must be contained within apostrophes (single quotes). For example:
 

```
//V71AWLM1 PROC   DB2SSN-V71A, NUMTCB=1, APPLENV='WLM_ENV1'
//TCBNUM1 EXEC   PGM=DSNX9WLM, TIME=1440
//                PARM='&DB2SSN,&NUMTCB,&APPLENV'
//                REGION=OM
```

The second method does not give you a migration path because your DSNTIDxx member and DSNTIJUZ parameters are not saved for future input. In addition, this

second method does not generate the sample jobs DSNTSJ6S, DSNTSJ6P, DSNTSJ6D, DSNTSJ6T, DSNTSJ6U, DSNTSJ6V, DSNTSJ6W, DSNTSJ6Z, DSNTSJ61, DSNTSJ62, DSNTSJ63, DSNTSJ64, and DSNTSJ65 because you are not running the DSNTINST CLIST.

For more information on stored procedures, see Part 6 of *DB2 Application Programming and SQL Guide*.

## Enabling DB2-supplied stored procedures

You can enable the DB2-supplied stored procedures after you have completed the installation process. The DB2-supplied stored procedures are:

<b>DSNACCAV</b>	DB2 UDB Control Center partition information stored procedure
<b>DSNACCDD</b>	DB2 UDB Control Center stored procedure to delete data sets
<b>DSNACCDE</b>	DB2 UDB Control Center stored procedure to indicate if a data set exists
<b>DSNACCDL</b>	DB2 UDB Control Center stored procedure to list data sets
<b>DSNACCDR</b>	DB2 UDB Control Center stored procedure to rename data sets
<b>DSNACCDs</b>	DB2 UDB Control Center stored procedure to create, append to, or replace LCRECL=80, RECFM=FB PDSE data set members or PS datasets
<b>DSNACCQC</b>	DB2 UDB Control Center catalog query stored procedure
<b>DSNACICS</b>	CICS transaction invocation stored procedure that invokes CICS server programs
<b>DSNTPSMP</b>	SQL procedures processor stored procedure
<b>DSNUTILS</b>	Utility invocation stored procedure
<b>DSNWZP</b>	Subsystem parameter stored procedure, which is used by DB2 Visual Explain and WLM_REFRESH
<b>INSTALL_JAR</b>	Stored procedure that installs a set of Java classes into the current SQL catalog and schema
<b>REMOVE_JAR</b>	Stored procedure that removes a Java jar file and its classes from a specified catalog
<b>REPLACE_JAR</b>	Stored procedure that replaces a previously installed jar

For more information on stored procedures, see Part 6 of *DB2 Application Programming and SQL Guide*. See Appendix B of *DB2 Utility Guide and Reference* for information on stored procedures that perform utility functions.

## Enabling Control Center procedures

If you have already installed the DB2 UDB Control Center FMID, JDB771D, use job DSNTIJCC to define the DB2-supplied stored procedures for DB2 UDB Control Center. Installation job DSNTIJCC creates a database that is used by CC/390 to contain information about templates, object lists, and utility procedures. The database also contains information that the DB2 UDB Control Center uses to restart stopped utilities. By default, installation job DSNTIJCC names this database CC390.

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 Part 6 of *DB2 Application Programming and SQL Guide*.

## Enabling SQL procedure support

You can enable SQL procedures support after you have completed the installation process. The objects required for the DB2 SQL procedures support are:

- DSNHSQL is the JCL procedure for preparing SQL procedures in batch mode.
- DSNTPSMP is a stored procedure that can be used to prepare SQL procedures for execution.
- The DSNTPSMP REXX EXEC is in the *prefix*.NEW.SDSNCLST data set.

If you use the SQL procedures processor, you need a WLM start-up procedure and the DB2 REXX Language Support feature. Copy the WLM start-up procedure, DSN8WLMP, from DSN710.SDSNSAMP and customize it.

## # Enabling CICS transaction invocation procedure

# DSNACICS needs to run in a WLM-established stored procedure address space.  
# Although DSNACICS can run in an address space with other stored procedures,  
# DSNACICS performs better if it runs in its own address space. Therefore, it is  
# recommended that you set up a WLM application environment and startup  
# procedure specifically for running DSNACICS.

# After you have set up the WLM application environment, you need to create a JCL  
# startup procedure for the stored procedure address space.

# Installation job DSNTIJMV contains a sample startup procedure for a stored  
# procedure address space for running DSNACICS. The procedure is named  
# DSNACICS. When you run the installation or migration CLIST, DB2 customizes  
# DSNTIJMV with data set names that you specify. Running DSNTIJMV installs the  
# startup procedure in your SYS1.PROCLIB data set. Use panel DSNTIPW to specify  
# CICS data set names. CICS EXCI LIBRARY is a new field in which you specify the  
# library that contains the load modules for the CICS EXCI interface, which  
# DSNACICS uses to connect to CICS and call CICS programs.

# For information on how to set up a WLM environment and associate an address  
# space startup procedure with that environment, see OS/390 MVS Planning:  
# Workload Management and DB2 Administration Guide.





---

## Chapter 8. Migrating from Version 5 to Version 7

This chapter describes the steps necessary to migrate from DB2 Version 5.

Before you begin migration, you need to load the DB2 libraries. This process is described beginning on page 51. You also need to run the installation CLIST. The process is described beginning on page 77. Also see “Migration Considerations” for information on changes that might affect your migration process.

When you migrate to Version 7, 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 Version 5, unpredictable results can occur after migration.

Make sure that your Version 5 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.

Before migrating the first member to Version 7, check the service levels of any coupling facilities running with coupling facility control code (CFCC) at CFLEVEL=7 or CFLEVEL=8. Coupling facilities at CFLEVEL=7 should have service level 1.06 or higher, and coupling facilities at CFLEVEL=8 should have service level 1.03 or higher. If these service levels are not installed, data corruption can occur. No service level requirements exist for the other CFLEVELs.

You can use the MVS D CF command to display the service level of coupling facilities. To use this command with releases of OS/390 previous to Version 2 Release 10, APAR OW38667 must be applied.

You must use primary authorization IDs to perform the following migration steps.

---

### Migration Considerations

Be aware of the following changes that might affect your migration to Version 7. For information on how migration might affect your DB2 operations, see “Make adjustments for release incompatibilities” on page 281.

#### Scrollable cursors are supported

Support for scrollable cursors enables random access to data in a table. Temporary result tables are stored in declared temporary tables. You must allow sufficient storage space for the new TEMP database and segmented table spaces.

#### Unicode support

If you plan to use Unicode, all members of a data-sharing group must convert to Version 7.

#### Encoding schemes of string parameters for routines

The new PARAMETER CCSID clause allows you to define the encoding scheme of all string parameters for stored procedures and system-defined parameters at once. In previous versions, you had to define a CCSID for each string parameter if you

wanted an encoding scheme other than the default. Also, EBCDIC is no longer the default encoding scheme for system-defined parameters. DB2 now uses the same encoding scheme for both user-specified and system-generated string parameters.

## **Revoking SYSADM authorization**

If an authorization ID with SYSADM authority grants USAGE on a JAR (Java ARchive) in Version 7, an attempt to revoke SYSADM from this ID in a previous release will fail and message DSNT501I will be issued. SYSADM authority can be revoked only from Version 7.

## **Changed messages from the LOAD utility**

In previous releases of DB2, if the data types of input data and the column into which the data was loaded were incompatible, DB2 issued message DSNU390I during the UTILITY phase of utility processing and terminated the utility with return code 8. In Version 7, if the data types are incompatible, DB2 issues message DSNU299I during the RELOAD phase. If LOAD is not performing discard processing, LOAD abnormally terminates with reason code 00E40323. If LOAD is performing discard processing, LOAD issues message DSNU299I for every row of input data and puts the rejected records in the discard file. LOAD then terminates with return code 8.

## **Change in recording soft errors in SYS1.LOGREC data set**

In previous versions of DB2, abends that occur as a result of errors in SQL statements were handled by DB2 recovery routines and were recorded in SYS1.LOGREC. Examples of these errors include:

- An error in decimal arithmetic
- Overflow error
- Underflow error

In Version 7, by default, these abends are not recorded in SYS1.LOGREC. If you want these errors to be recorded in SYS1.LOGREC, you can specify YES in the RECORD SOFT ERRORS field of installation panel DSNTIPM.

## **More than 32 K databases are supported**

The maximum number of databases is no longer 32 K. The database identifier column of catalog table SYSIBM.SYSDATABASE can contain negative numbers, to indicate that there are more than 32 K databases defined.

## **Support for up to 150 000 connections**

To support up to 150 000 connections, the token for displaying LUWIDs for DISPLAY THREAD is now a 6-digit decimal number.

## **Log buffer size increased**

The maximum log output buffer size is 100 000 4 KB buffers (400 megabytes). The input read buffer size is increased to 60 KB. You will probably experience better log read and log write performance with these increases.

## **Increased maximum number of data sets open**

The maximum number of concurrently allocated data sets increases to 32 767 for customers running OS/390 Version 2 Release 6. The practical limit for concurrently

allocated data sets depends on virtual storage below the 16M line, OS/390 allocation control blocks, and some DB2 storage. For more details, refer to Part 5 (Volume 2) of *DB2 Administration Guide*.

## Changes to the RLST

The RLST has new optional columns and a new search order for the index. You can use an RLST that has been defined from a release earlier than Version 7 (the RLST index is in ascending order, and there are no new columns), but you cannot do predictive governing, and you cannot take advantage of the performance improvements with the new descending index. You can alter an existing RLST to add the new columns and change the definition of the index. Migration job DSNTIJSJG modifies your existing RLST.

## Creating tables with DBCS and mixed columns

You can no longer create extended binary-coded decimal interchange code (EBCDIC) tables with GRAPHIC, VARGRAPHIC, DBCLOB, or CHAR FOR MIXED DATA columns or alter EBCDIC tables to add GRAPHIC, VARGRAPHIC, DBCLOB or CHAR FOR MIXED DATA columns when the setting for installation option MIXED DATA is NO.

## Increased size for generated code

In DB2 Version 7, you can specify CCSIDs for host variables. Because DB2 stores the CCSIDs with the host variables, the generated source code and object code for a DB2 application might be larger than in previous versions.

## Consider increasing IDBACK and CTHREAD

Because utilities may use additional threads, you should consider increasing the IDBACK and CTHREAD subsystem parameters. Increasing these parameters can help you avoid failure of some utilities due to increased thread usage. An increase also supports the additional parallelism to be performed by the utilities.

## Customized DB2I defaults can be migrated

A DB2I TSO IPSF profile member from a prior release can be migrated to the current release. The DSNEMC01 CLIST uses the values specified on installation panel DSNTIPF and stores the results in the ISPF profile member DSNEPROF. Any customized DSNEPROF members can be migrated from Version 5 to Version 7. However, you need to examine any new or changed default panel values to ensure that your customized values are still valid.

## Migrating 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 3 of *DB2 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 Version 5 load library for each member in the data sharing group. For best availability, you can apply the SPE to one member at a time. You can have Version 5 systems with and without the SPE running at the same time. Stop and restart each member to pick up the change. Before migrating the first member of the data sharing group to Version 7, the fallback SPE must be applied to all members in the group.

4. Follow the procedure on migrating the data sharing group in Chapter 4 of *DB2 Data Sharing: Planning and Administration*. You must complete the migration of the first member of the data sharing group to Version 7 before starting any other members at Version 7.
5. To prepare for fallback, keep the subsystem parameter load module used by Version 5.

The CLIST edits different jobs for enabling data sharing and migrating a data sharing member. See Chapter 4 of *DB2 Data Sharing: Planning and Administration* for the list of jobs that are edited for each case.

## Consider enlarging BSDS

The BSDS may need to be larger to accommodate the additional buffer pools. To avoid the BSDS going into secondary extents, change the record size of the primary allocation to 180 records. To increase the space allocation for the BSDS, you must:

1. Rename existing BSDSs.
2. Define larger BSDSs with the original names.
3. Copy the renamed ones into the new ones.

You can do this using access method services. See the Version 7 installation job DSNTIJIN for the definition using the larger primary extent size.

## Stored procedures

In earlier releases of DB2, columns AUTHID and LUNAME of catalog table SYSIBM.SYSPROCEDURES were used to uniquely identify multiple instances of a procedure. After migrating to Version 7, use the SCHEMA column in the SYSIBM.SYSROUTINES catalog table, the CURRENT PATH special register, and the PATH bind option to identify multiple instances of a procedure. During migration, DB2 generates CREATE PROCEDURE statements which populate SYSIBM.SYSROUTINES and SYSIBM.SYSPARMS. Rows in SYSIBM.SYSROUTINES that contain non-blank values for columns AUTHID or LUNAME are not used to generate the CREATE PROCEDURE statements. You can identify those rows by using the following statement:

```
SELECT * FROM SYSIBM.SYSPROCEDURES WHERE AUTHID <> ' ' OR LUNAME <> ' ';
```

DB2 also copies rows in SYSIBM.SYSPROCEDURES into table SYSIBM.SYSPARMS and extends information from the PARMLIST column of SYSIBM.SYSPROCEDURES into SYSIBM.SYSROUTINES and SYSIBM.SYSPARMS.

Procedures migrated from Version 5 do not have an owner, because they were not created with the CREATE PROCEDURE statement. DB2 authorization treats these migrated procedures differently than procedures created with CREATE PROCEDURE. The authorization for migrated procedures is unchanged.

Tables SYSIBM.SYSPSM and SYSIBM.SYSPSMOPTS are not used in Version 7. You have these tables if you are currently using Version 5 SQL Procedures support. Job DSNTIJMP migrates the data from these tables into new catalog tables. For more information, see "Migration step 22: Migrate SQL procedures tables: DSNTIJMP (optional)" on page 302.

Catalog table SYSIBM.SYSPROCEDURES is not used after Version 5 to define stored procedures to DB2. The SYSCAT.PROCEDURES view of catalog table

SYSIBM.SYSPROCEDURES is not used to list the stored procedures registered in DB2. Although these tables are no longer used in Version 7, DB2 does not drop them during migration because they are needed for data sharing coexistence and fallback.

For more information, refer to *DB2 Application Programming and SQL Guide*.

## Utility enhancements

The online REORG performance has been enhanced. The performance improvements are explained in Part 5 (Volume 2) of *DB2 Administration Guide*. The enhancements use new calculations for the sort and unload data sets. You may need to increase the size of your unload and sort data sets. The new calculations are explained in *DB2 Utility Guide and Reference*.

## Work file database size calculations

The migration job DSNTIJTC creates and updates indexes on catalog tables. These indexes are created and updated *sequentially* during migration. The work file database is used for the sort of each index; DB2 needs enough work file storage to sort the largest of the indexes in Table 52. The migration fails if you do not have enough storage, so ensure that you have enough space before you begin. See “Disk requirements for the work file database” on page 29 for information about space requirements.

Table 52 shows the indexes that are new and changed for existing catalog tables.

Table 52. Indexes added or updated sequentially using the work file database

Catalog table name	Index name	Column names
SYSIBM.SYSINDEXPART	SYSIBM.DSNDRX02	STORNAME
SYSIBM.SYSCOLUMNS	SYSIBM.DSNDCX02	TYPESHEMA, TYPENAME
SYSIBM.SYSPACKDEP	SYSIBM.DSNKDX03	BQUALIFIER, BNAME, BTYPE, DTYPE
SYSIBM.SYSTABLEPART	SYSIBM.DSNDPX02	STORNAME
SYSIBM.SYSVIEWDEP	SYSIBM.DSNGGX03	BSHEMA, BNAME, BTYPE
SYSIBM.SYSTABAUTH	SYSIBM.DSNATX02	GRANTEE, TCREATOR, TTNAME, GRANTEETYPE, UPDATECOLS, ALTERAUTH, DELETEAUTH, INDEXAUTH, INSERTAUTH, SELECTAUTH, UPDATEAUTH, CAPTUREAUTH, REFERENCESAUTH, REFCOLS, TRIGGERAUTH
SYSIBM.SYSUSERAUTH	SYSIBM.DSNAUH01	GRANTEE, GRANTEDTS
SYSIBM.SYSINDEXES	SYSIBM.DSNDXX02	DBNAME, INDEXSPACE, COPY, COPYLRN
SYSIBM.SYSVIEWS	SYSIBM.DSNVFX01	CREATOR, NAME, SEQNO, TYPE
SYSIBM.SYSTRIGGERS	SYSIBM.DSNOTX03	SCHEMA, TRIGNAME
SYSIBM.SYSROUTINES	SYSIBM.DSNOFX07	NAME, PARM_COUNT, ROUTINETYPE, SCHEMA, PARM_SIGNATURE, PARM1, PARM2, PARM3, PARM4, PARM5, PARM6, PARM7, PARM8, PARM9, PARM10, PARM11, PARM12, PARM13, PARM14, PARM15, PARM16, PARM17, PARM18, PARM19, PARM20, PARM21, PARM22, PARM23, PARM24, PARM25, PARM26, PARM27, PARM28, PARM29, PARM30
SYSIBM.SYSROUTINES	SYSIBM.DSNOFX08	JARSHEMA, JAR_ID

## Data set password protection is removed

Remove data set passwords before migrating to Version 7. Use OS/390 Security Server or an equivalent security system to protect your data sets. Catalog migration

will issue a warning message if any data set passwords are found. See “Identify unsupported objects (DSNTIJPM)” on page 279 for more information.

## **SYSIBM.SYSPROCEDURES no longer used**

Catalog table SYSIBM.SYSPROCEDURES is no longer used to define stored procedures to DB2. During catalog migration, all rows in SYSIBM.SYSPROCEDURES are automatically migrated to the new SYSIBM.SYSROUTINES and SYSIBM.SYSPARMS catalog tables.

In Version 7, you use the CREATE and ALTER PROCEDURE statement to define stored procedures. To prepare for fallback, or to run in a coexistence environment for data sharing, you must modify the SYSPROCEDURES catalog table to match the updates to SYSROUTINES and SYSPARMS that those CREATE and ALTER statements make.

## **Private protocol function not enhanced**

No enhancements are planned for distributed data using private protocol. Private protocol support may be removed in a later release of DB2. Private protocol cannot use type 2 inactive threads. Specify a non-zero value for MAXTYPE1 to use type 1 inactive threads for private protocol. To take advantage of the enhancements to stored procedures, TCP/IP and new data types, you must use the DRDA protocol. You can use DRDA without having to change your applications by rebinding with the DBPROTOCOL option set to DRDA. See Chapter 2 of *DB2 Command Reference* for more information.

## **Type 2 indexes are required**

There is no longer support for type 1 indexes. You must convert all indexes to type 2 before migrating to Version 7. If any type 1 indexes are found, catalog migration will fail. See “Identify unsupported objects (DSNTIJPM)” on page 279 for more information.

## **Shared read-only data is removed**

Data sharing is a more substantial and more usable function than shared read-only data. You can also use distributed data to share information. If any shared read-only data is found during catalog migration, you will receive a warning message. See “Identify unsupported objects (DSNTIJPM)” on page 279 for more information about eliminating shared read-only data.

## **DCE security authentication no longer supported**

DCE support has been removed. Kerberos authentication has replaced DCE. For more information about creating Kerberos RACF profiles, see Part 3 (Volume 1) of *DB2 Administration Guide*.

## **Release coexistence**

This section highlights considerations for coexistence between a previous release and Version 7 in a data sharing and in a distributed environment.

### **IRLM**

As you apply IRLM service to members of a data sharing group, some members run with the newer service level and some with the older service level. Even though each member uses the same release level of IRLM, the different service levels can raise issues you must consider. For more information about IRLM coexistence, see *DB2 Data Sharing: Planning and Administration*.

## Data sharing

DB2 supports both Version 6 and Version 7 or both Version 5 and Version 7 members in a data sharing group, but not all three versions at once. DB2 will support the coexistence of only two releases at a time. To support two releases, you must first apply the fallback SPE to all Version 5 members of the group as described in “Migrating a data sharing group” on page 273. Release coexistence begins when you migrate the first data sharing member to Version 7. You must successfully migrate the first data sharing member to Version 7 before attempting to migrate the other data sharing members.

For the best availability, you can migrate the members to Version 7 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, **or**
- Restrict the execution of packages and plans bound on Version 7 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 Structured Query Language (SQL) statement that is not allowed in Version 5. Secondly, you avoid the automatic rebind that occurs when any plan or package that is bound on Version 7 is run on Version 5. It also avoids the automatic rebind that occurs when a Version 7-bound plan or package that was automatically rebound on Version 5 is later run on Version 7.

If it is not possible to enforce on which member a plan or package runs, consider how to handle binds and automatic rebinds while two releases are coexisting. For more information, see the description of the AUTOBIND option on installation panel DSNTIPO.

For detailed information about data sharing release coexistence considerations, see *DB2 Data Sharing: Planning and Administration*.

**TSO and CAF logon procedures:** 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. If you forget to update those procedures and jobs before migrating to any release subsequent to Version 7, those procedures and jobs will no longer work in that subsequent release.

For a detailed list of considerations for a data sharing group with multiple DB2 releases, see Chapter 4 of *DB2 Data Sharing: Planning and Administration*.

## Distributed environment

DB2 for OS/390 and z/OS 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 and z/OS can be used only when using DRDA access.

Other DRDA partners at DRDA level 4 can also take advantage of the functions introduced in this release of DB2 for OS/390 and z/OS.

---

## Migration step 1: Premigration activities

This step is included as a reminder to do the following tasks before migrating to Version 7:

- “Remove incomplete table definitions”
- “Identify unsupported objects (DSNTIJPM)” on page 279
- “Save critical access paths (optional)” on page 280
- “DRDA support for three-part names (optional)” on page 281
- “Make adjustments for release incompatibilities” on page 281

### Remove incomplete table definitions

Before migrating to Version 7, you must remove all incomplete table definitions. The following query identifies existing incomplete table definitions that must be fixed:

```
SELECT  COLNO, TBCreator, TBNAME, REMARKS
FROM    SYSIBM.SYSCOLUMNS
WHERE   COLNO = 0;
```

There are several ways to resolve incomplete table definitions:

- Drop the table if it is not needed.
- Complete the definition.

The REMARKS field for each table identified in the SELECT result lists the column numbers of unique key constraints that do not have enforcing indexes. Commas separate constraint column sets, and spaces separate column numbers with constraints. Create a unique index for each constraint that is listed in the incomplete table definition. For example:

COLNO	TBCreator	TBNAME	REMARKS
0	SYSADM	TBTEST1	4 12 23,5 9 relname,11

There are three constraints that do not have enforcing indexes. One of the constraints is a referential constraint with a constraint name of “relname”. The referential constraint name can be ignored because it is not needed for the purpose of completing the table definition. The following is an example of a SELECT statement to identify the column names corresponding to the column number in the REMARKS field:

```
SELECT  TBCreator, TBNAME, COLNO, NAME
FROM    SYSIBM.SYSCOLUMNS
WHERE   TBCreator = 'SYSADM'
AND     TBNAME = 'TBTEST1'
AND     COLNO IN ( 4, 12, 23, 5, 9, 11 );
```

Create indexes by using the column numbers from the SELECT results:

```
CREATE UNIQUE INDEX IXTEST1
ON SYSADM.TBTEST1( COL4, COL12, COL23 );
CREATE UNIQUE INDEX IXTEST2
ON SYSADM.TBTEST1( COL5, COL9 );
CREATE UNIQUE INDEX IXTEST3
ON SYSADM.TBTEST1( COL11 );
```

Creating the index may not remove the incomplete table definition. In this case, use the third method below to remove the incomplete table definitions.



- Proceed with migration and re-create the table. After migration, unload all data from the table, drop the table, and re-create the table with the constraints and indexes.

During migration to Version 7, DB2 will search SYSCOLUMNS for incomplete table definitions. If any are found, a warning message will be issued.

## Identify unsupported objects (DSNTIJP)

Version 7 does not support the following items. If you do not remove the following items from your catalog, you will receive a warning message during migration, and the items will not be usable:

- Type 1 indexes on the catalog or directory (migration will fail)
- Data set passwords for all objects except the bootstrap data set (BSDS)
- Shared read-only data
- Views on SYSIBM.SYSCOLDIST or SYSIBM.SYSCOLDISTSTATS

There are several ways to identify these objects:

- Run job DSNTIJP. This job queries the catalog and identifies objects that are no longer supported in Version 7. The job generates SQL ALTER statements and REBUILD index statements that you can use to change the objects. This job is found in *prefix.SDSNSAMP*.
- Run your own catalog queries. The following queries are also included in member DSNTESQ in the sample library *prefix.SDSNSAMP*:

```
SELECT * FROM SYSIBM.SYSINDEXES
      WHERE INDEXTYPE <> '2' ;

SELECT NAME, BPOOL, ROSHARE FROM SYSIBM.SYSDATABASE
      WHERE ROSHARE <> ' ' ;

SELECT CREATOR, NAME
      FROM SYSIBM.SYSINDEXES
      WHERE DSETPASS <> ' ' ;

SELECT DBNAME, NAME
      FROM SYSIBM.SYSTABLESPACE
      WHERE DSETPASS <> ' ' ;
SELECT DCREATOR, DNAME
      FROM SYSIBM.SYSVIEWDEP
      WHERE BCREATOR = 'SYSIBM'
      AND BNAME IN( 'SYSCOLDIST','SYSCOLDISTSTATS' );
      AND BNAME IN( SELECT TBNAME
                     FROM SYSIBM.SYSCOLUMNS
                     WHERE TBcreator = 'SYSIBM'
                     AND TBNAME IN( 'SYSCOLDIST', 'SYSCOLDISTSTATS' )
                     AND NAME = 'COLVALUE'
                     AND LENGTH <> 255 );
```

## Convert indexes to type 2

Because type 2 indexes often take more space than type 1, first determine if you need to allocate more space for the indexes. Refer to Section 1 of *DB2 Administration Guide* for more information. Based on these calculations, if your index space is not large enough, delete and redefine your data sets with a larger allocation. For more information on increasing your data set allocations, see *DFSMS/MVS: Storage Administration Reference for DFSMSdfp*.

You can convert indexes to type 2 as follows:

- Use the CONVERT TO TYPE 2 option of the ALTER INDEX SQL statement. This method works only with catalog indexes, not directory indexes.

1. Enter the SQL statement ALTER INDEX with the CONVERT TO TYPE 2 option.
2. Run the utility REBUILD INDEX. See *DB2 Utility Guide and Reference* for more information on REBUILD INDEX.

**Recommendation:** Run REBUILD INDEX on an index immediately after running ALTER INDEX because the index is unusable until it is rebuilt.

- Run the CATMAINT utility with the CONVERT TO TYPE 2 option. This will convert catalog or directory indexes to type 2. See *DB2 Utility Guide and Reference* for more information on CATMAINT.

### Remove data set passwords

To remove passwords from your data sets, use the ALTER statement to change the password to a blank. (If you ran DSNTIJPM, this job generated the ALTER statements for you.)

Use OS/390 Security Server or an equivalent to protect access to your data sets.

### Eliminate shared read-only data

Use the following steps to eliminate shared read-only data:

1. Enter the SQL statement DROP DATABASE for all databases defined as ROSHARE READ.
2. For databases defined as ROSHARE OWNER, enter the ALTER DATABASE statement with the ROSHARE NONE option.

### Remove views on two catalog tables

Before running job DSNTIJTC (CATMAINT), remove all views on catalog tables SYSIBM.SYSCOLDIST and SYSIBM.SYSCOLDISTSTATS. Catalog migration will fail if any views are found. After you have successfully migrated your catalog to Version 7, you can redefine the views. Run the following query to determine whether you have these views:

```
SELECT DCREATOR, DNAME
FROM SYSIBM.SYSVIEWDEP
WHERE BCREATOR = 'SYSIBM'
AND BNAME IN ( 'SYSCOLDIST', 'SYSCOLDISTSTATS' )
AND BNAME IN ( SELECT TBNAME
                FROM SYSIBM.SYSCOLUMNS
                WHERE TBCREATOR = 'SYSIBM'
                AND TBNAME IN( 'SYSCOLDIST', 'SYSCOLDISTSTATS' )
                AND NAME = 'COLVALUE'
                AND LENGTH <> 255 );
```

If any rows are returned from the above query, you must drop the views before migrating to Version 7.

## Save critical access paths (optional)

Sometimes changes between releases of DB2 cause unwanted access path changes. Consult with your performance analysts to determine which queries are especially critical and ensure that a PLAN\_TABLE exists that contains the good access path.

EXPLAIN your queries before migrating. Because EXPLAIN requires a rebind, this could change your access paths. Therefore, extract the needed queries and then EXPLAIN them under a different application or program name. This protects the existing application while the access path information is obtained. Then, after the access paths for the extracted queries are validated, you can update the

APPLNAME or PROGNAME columns of PLAN\_TABLE to the correct name. For more information about using access path hints, see Part 5 (Volume 2) of *DB2 Administration Guide*.

To make sure that you do not lose any rows in the current PLAN\_TABLE, use ALTER TABLE to add the new Version 7 columns to the existing PLAN\_TABLE. See Chapter 5 of *DB2 SQL Reference*.

## DRDA support for three-part names (optional)

A new bind parameter, DBPROTOCOL, enables application programs that use three-part names for remote access to use DRDA protocol. Existing applications can be converted from private protocol to DRDA with minimal rework. Packages need to be rebound with the DBPROTOCOL parameter. The governing row in the RLST at the server needs to be changed from a row that governs by plan to one that governs by package.

## Examine all new and changed values for DB2I panels

The DB2I default panels DSNEOP01 and DSNEOP02, will not be initialized with the values specified during the installation CLIST process during a migration. The DB2I panel variables in the ISPF profile from the previous release will be used on the current release. Any customized DSNEPROF members will be migrated from Version 5 to Version 7. However, you must examine any new or changed default panel values to ensure that your customized values are still valid.

## Make adjustments for release incompatibilities

These changes might affect your DB2 operations after migrating to Version 7.

### Adjust application programs

You might need to adjust your application programs because of the following release incompatibilities.

**SQLCODE -101:** After migrating to DB2 Version 7, 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.

Rewrite the unsuccessful SQL statements by using correlated references, breaking up UNIONS or using OUTER JOINS.

**SQLCODE -538:** After migrating to DB2 Version 7, it is possible to receive SQLCODE -538 on an SQL CREATE TABLE FOREIGN KEY or SQL ALTER TABLE ADD FOREIGN KEY statement that previously executed successfully. If the foreign key references a parent key that has not been defined as a primary key or unique key in the parent table, the foreign key creation will fail. You must ensure that the parent key has been created as a primary key or unique key in the parent table.

**SQLCODE -669:** After migrating to DB2 Version 7, it is possible to receive SQLCODE -669 on an SQL DROP INDEX statement that previously executed successfully. If the dropped index is a unique index used to enforce a unique (primary or unique key) constraint or referential constraint, you must drop the constraint before you can drop the index.

It is possible to drop a unique index (for a unique key only) without first dropping the unique key constraint if the unique key was created in a release of DB2 prior to Version 7, and the unique key has no associated referential constraints.

**Change applications that call DSNACCQC or DSNACCAV:** You may need to change applications that call DB2-supplied stored procedures DSNACCQC or DSNACCAV. The stored procedures are now defined to run in a WLM-established stored procedures address space. The single result set that is returned by DSNACCQC and the second result set that is returned by DSNACCAV contain one more column than in previous releases of DB2.

**Changed encoding scheme inheritance:** Any CREATE TABLE, CREATE GLOBAL TEMPORARY TABLE, and DECLARE GLOBAL TEMPORARY TABLE statements with a LIKE clause, but no CCSID and IN clauses, will inherit the encoding scheme of the table being copied. Previously, these tables would have inherited the system default encoding scheme.

**No colon on a host variable is an error:** All host variable references must have the leading colon. If you neglect to use a colon to designate a host variable, the precompiler issues a DSNH104I message or interprets the host variable as an unqualified column name, which might lead to unintended results. The host variable without a colon is interpreted as a column name when the host variable is referenced in a context in which a column name can also be referenced.

To find out if you have host variables that are not preceded with a colon, precompile again in Version 5 and check for warning message DSNH315I. You can correct the problem by inserting colons in the source code, precompiling, and binding the new DBRM. If you do not have source code, read APARs II12100, PQ26922, and PQ30390. For more information about detecting missing colons on host variables and correcting the problem, see DB2 UDB Server for OS/390 Version 6 Technical Update, SG24-6108-00.

**Changed format for DSN messages:** The DSN message identifiers can now be 8 or 9 characters long. The message identifier formats are DSNcnnnl and DSNcnnnnl. To accommodate the longer message identifier, the message text will begin in column 13 for all DSN messages. If you have applications that scan the text of DSN messages, you might need to modify them.

**Changed format for message DSNU050I:** Certain formatting problems in message DSNU050I have been corrected. The message formats differently in Version 7 than it did in Version 5. If you have applications that scan the text of message DSNU050I, you might need to modify them.

**SQL reserved words:** There are several new SQL reserved words in Version 7. Refer to *DB2 SQL Reference* for the list and adjust your applications accordingly.

**Using the euro symbol:** New CCSIDs that support the euro symbol have been added to DB2. See "Appendix A. Character conversion" on page 523 for details on the coded character set identifiers.

**Using aliases:** Prior to Version 7, aliases were known locally. Now, if remote aliases are used in SQL, they should be defined at the server site. With SET statements that assign a host variable with the value of a special register, the host variable is only assigned the local value.

**QUIESCE return code:** The QUIESCE utility command produces a return code of 4 for duplicate names in the list of table spaces to be quiesced. This is a change from a return code of 8. The QUIESCE processing continues and the table space is quiesced only once.

**DSNH message ID lengths:** DSNH messages can be 8, 9 or 10 characters long. Existing messages in Version 7 will not be padded with zeroes to 10 characters.

**Positive SQLCODE from PREPARE:** A return code greater than zero can be returned from a PREPARE statement if your migrated application is using predictive governing or optimization hints.

**New DBPROTOCOL option:** Applications may be affected by the new DBPROTOCOL option which is both a subsystem parameter and a bind parameter. If the DBPROTOCOL option is not coded on the BIND, the value listed in the subsystem parameter DBPROTCL will be used. For plans and packages to execute the same as they did in Version 5, they may need to be rebound depending on the DBPROTOCOL value selected. In prior releases, remote access by name resulted in the use of DB2 Private Protocol to communicate with a remote DB2 for OS/390 and z/OS site. Remote access with the CONNECT statement used the DRDA database protocol. Users can now specify which protocol to use when communicating with the remote server.

**Subsystem parameter OJPERFEH not used:** Subsystem parameter OJPERFEH is no longer used.

**Changed default for RELCURHL subsystem parameter:** Use the RELCURHL subsystem parameter to indicate whether you want DB2 to release a data page or row lock at COMMIT for the row on which a cursor defined WITH HOLD is positioned. In Version 5, the default value was NO. Although this particular row or page lock is not necessary for maintaining cursor position, NO was provided as a default for existing applications that relied on that data lock. In Version 7, RELCURHL is a field on installation panel DSNTIP4. The field name is RELEASE LOCKS, and the default is changed to YES to provide for better concurrency (see 171 for more information). If you still require the lock, specify NO for RELEASE LOCKS.

**Changed default for DYNRULS subsystem parameter:** In earlier releases, the default for the DYNRULS subsystem parameter is NO. This means that the values of the precompiler options are used for dynamic SQL statements in plans or packages bound with DYNAMICRULES(BIND). For Version 7, the default is YES. YES means the application programming (DSNHDECP) defaults will be used for dynamic SQL statements in plans or packages bound using DYNAMICRULES bind, define, or invoke behavior.

**Using new column called CLUSTERRATIOF:** The CLUSTERRATIOF column has been added to SYSINDEXES and SYSINDEXSTATS. After migration this column will have a default value until it is updated by RUNSTATS or some other application. If the optimizer sees the default value in CLUSTERRATIOF, it will use the value in CLUSTERRATIO.

**Support for large objects:** The following incompatibilities result from the expansion of the SQLDA to support large objects:

- Because of the additional data types supported after Version 5, the host language statements generated by the precompiler for each SQL statement include a larger parameter list. As a result, the size of the object code that results

- INCLUDE SQLDA generates additional fields for applications written in assembler, PL/I, C, and C++ (see Appendix C of *DB2 SQL Reference* for descriptions of these fields). If an SQL application program written in one of those languages defines a name that matches any symbol in the additional fields, and the program also uses INCLUDE SQLDA, then that program might not operate, assemble, or compile correctly.

**A technique for reducing the number of matching columns no longer works:**  
Part 5 (Volume 2) of *DB2 Administration Guide* formerly described a technique for reducing the number of matching columns for an index by changing equal predicates to BETWEEN predicates. Due to enhancements to BETWEEN predicate processing, this technique no longer reduces the number of matching columns. If you have written queries that use this technique, consider using the alternative technique of discouraging the use of a particular index.

***New reserved qualifier for tables, SESSION:*** DB2 uses a new implicit qualifier, SESSION for a declared temporary table. All tables with a high level qualifier of SESSION will be treated as a declared temporary table. You must modify any existing tables with a high level qualifier of SESSION to use another qualifier.

Use the following query to identify existing tables, views, aliases, and synonyms that will be inaccessible because of the 'SESSION' qualifier:

## Product-sensitive Programming Interface

```
SELECT 'TABLE', CREATOR, NAME
  FROM SYSIBM.SYSTABLES
 WHERE CREATOR = 'SESSION' AND TYPE IN ('T', 'G', 'X')
UNION
SELECT 'VIEW', CREATOR, NAME
  FROM SYSIBM.SYSTABLES
 WHERE CREATOR = 'SESSION' AND TYPE = 'V'
UNION
SELECT 'ALIAS', CREATOR, NAME
  FROM SYSIBM.SYSTABLES
 WHERE CREATOR = 'SESSION' AND TYPE = 'A'
UNION
SELECT 'SYNONYM', CREATOR, NAME
  FROM SYSIBM.SYSSYNONYMS
 WHERE CREATOR = 'SESSION';
```

**End of Product-sensitive Programming Interface**

DB2 cannot know during the bind of the plan or package whether the static SQL statement that references a table name qualified by SESSION is for a persistent base table or for a declared temporary table. These static SQL statements will be incrementally bound at run-time when the static SQL statement is issued.

## Fewer sort operations for queries

A performance enhancement to DB2 results in fewer sort operations for queries that have ORDER BY clauses and have WHERE clauses with predicates of the form COL=constant. This enhancement can result in the following changes from previous releases of DB2:

- Changes in access paths, which can result in changes to EXPLAIN output or the elapsed time of a query.

- Some queries that previously received SQL code -136 (sort key length exceeds maximum) may now execute successfully.

### Sysplex query parallelism in the PLAN\_TABLE

If a parallel group being rebound to Version 7 is capable of sysplex query parallelism, and only one DB2 data sharing member is active, a single-CEC parallel plan is developed, and an **X** is placed in column PARALLEL\_MODE of table PLAN\_TABLE instead of the **C** that was used in Version 5. If you do not want the **X**, bind the plan or package on a member that is installed with COORDINATOR=NO. The **X** for column PARALLEL\_MODE means that all available assisting DB2s are used at run time. You need to rebind all static plans that have a PARALLEL\_MODE of X after migrating to take advantage of this change.

### Limit backouts with system restarts

There is a release incompatibility by using the AUTO setting of LIMIT BACKOUT, selected from installation panel DSNTIPL. The purpose of the setting is so that you can indicate that you want DB2 to postpone some of the backout work that is usually performed during system restart. **Recommendation:** Use the default setting, AUTO, but be aware that after completing restart, some objects (table spaces, index spaces or partitions) might be unavailable until the automatically started process completes the backout work. In releases prior to Version 6, all of DB2 was unavailable until the backout work was completed.

### DISPLAY BUFFERPOOL changes

The DISPLAY BUFFERPOOL command syntax and output have changed. The new command provides more data sharing information required for understanding the level of inter-system sharing taking place. The DISPLAY BUFFERPOOL output no longer issues messages DSNB450I, DSNB451I, and DSNB452I. Several new messages with expanded information are generated by this command. For details on the changes see the DISPLAY BUFFERPOOL command in *DB2 Command Reference*.

### Index changes

Indexes that have been altered, rebuilt, reorged, or newly created can increase in size by one page per nonpartitioned index or one page per index partition. The leaf distribution column (LEAFDIST) value in SYSINDEXPART may increase for indexes that have been altered, rebuilt, reorged, or newly created.

### ALTER INDEX syntax

After Version 5, ALTER INDEX is more flexible in allowing multiple PART clauses to be specified in a single ALTER INDEX statement. Existing ALTER INDEX statements that specify partition-level options before the PART keyword have different results in Version 7 than in previous releases. In Version 7, when these options are specified before the PART keyword, they apply to all partitions in the index unless you override them with a PART specification.

For example, assume that you have the following statement to change the FREEPAGE value to 10 for partition 1:

```
ALTER INDEX ixname
  FREEPAGE 10
  PART 1;
```

In Version 7, this same statement changes the FREEPAGE value to 10 for all index partitions. If you want to obtain the same results for this statement as you did in previous releases, change the statement as follows:

```
ALTER INDEX ixname
  PART 1
  FREEPAGE 10;
```

For the ALTER INDEX statement syntax, see *DB2 SQL Reference*.

### **RECOVER INDEX becomes REBUILD INDEX**

Before migrating to Version 7, ensure that you have applied APAR PQ09842, which enables you to change your RECOVER INDEX jobs to use the new syntax REBUILD INDEX. In Version 7, RECOVER INDEX means to use a copy of the index and apply log records. Thus, your old RECOVER INDEX jobs are likely to fail. Change any existing RECOVER INDEX utility jobs to use REBUILD INDEX. See APAR PQ09842 for more information.

### **Work space formulas changed for utilities**

Due to the larger table spaces and indexes available, the record header for several utility work data sets is lengthened by several bytes. The changed record headers which are used in several formulas for estimating work data set size affects the following utilities:

- CHECK DATA
- CHECK INDEX
- LOAD
- REBUILD INDEX
- REORG

The new calculations are explained in *DB2 Utility Guide and Reference* under the topic for defining work data sets for each utility.

### **Default for FASTSWITCH parameter is YES**

For some utilities, the FASTSWITCH parameter has a default of YES, which results in the creation of J0001 data sets. Because some tools and other utilities require data set names to match, you may experience incompatibilities. The fifth-level qualifier can vary from I0001 to J0001. For more information about renaming data sets, see Part 2 (Volume 1) of DB2 Administration Guide.

### **Change to parameter in IRLMPROC startup procedure**

The GROUP parameter in the IRLMPROC startup procedure has been changed to IRLMGRP. Be sure to use the IRLMGRP parameter instead of the GROUP parameter to prevent JCL errors during IRLM startup.

#  
#  
#  
#  
#  
#

---

## **Migration Step 2: Run link checker on DB2 for OS/390 Version 5 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, 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 ignore this message. See description of DSN1985I in *DB2 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 Part 3 of *DB2 Utility Guide and Reference*. For more information on CHECK INDEX, see Part 2 of *DB2 Utility Guide and Reference*.

You should run the following query on your Version 5 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 Version 7 member DSNTESQ 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 3: Determine which plans and packages are invalid after migration (optional)

Migrating to Version 7 renders some plans and packages invalid. Find out which ones are invalid by running the following queries on your Version 5 subsystem:

---

#### Product-sensitive Programming Interface

```
SELECT DISTINCT DNAME
FROM SYSIBM.SYSPLANDEP
WHERE BNAME IN ('DSNATX02',
                'DSNAUH01',
                'DSNDXX02',
                'DSNVVX01',
                'DSNVTH01',
                'SYSCOLUMNS', 'DSNDCX01',
                'SYSPACKAGE', 'DSNKKX01', 'DSNKKX02',
                'SYSPLAN', 'DSNPPH01',
                'SYSINDEXPART', 'DSNDRX01',
                'SYSTABLEPART', 'DSNDPX01')
      AND BCREATOR = 'SYSIBM'
      AND BTYPE IN ('I', 'T')
ORDER BY DNAME;
```

---

#### End of Product-sensitive Programming Interface

---

### Product-sensitive Programming Interface

```
SELECT DISTINCT COLLID, NAME, VERSION
FROM SYSIBM.SYSPACKDEP, SYSIBM.SYSPACKAGE
WHERE BNAME IN ('DSNATX02',
                'DSNAUH01',
                'DSNDXX02',
                'DSNVVX01',
                'DSNVTH01',
                'SYSCOLUMNS', 'DSNDCX01',
                'SYSPACKAGE', 'DSNKKX01', 'DSNKKX02',
                'SYSPLAN', 'DSNPPH01',
                'SYSINDEXPART', 'DSNDRX01',
                'SYSTABLEPART', 'DSNDPX01')
AND LOCATION = ' '
AND BQUALIFIER = 'SYSIBM'
AND BTYPE IN ('I', 'T')
AND COLLID = DCOLLID
AND NAME = DNAME
AND CONTOKEN = DCONTOKEN
ORDER BY COLLID, NAME, VERSION;
```

---

### End of Product-sensitive Programming Interface

These two queries are commented out in the Version 7 member DSNTESQ of *prefix.SDSNSAMP*. You can execute these queries from SPUFI or from a dynamic SQL program like DSNTDP2.

After migration, you can explicitly rebind these plans and packages, or let DB2 rebind them automatically. See Part 4 of *DB2 Application Programming and SQL Guide* for suggestions on rebinding these plans and packages.

---

## Migration step 4: 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 DSNTDP2.

Before you run these queries, you should already have run the DSN1CHKR utility and the CHECK INDEX utility in migration step 2.

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 5: Image copy directory and catalog in case of fallback: DSNTIJIC

### Attention:

You need to create a copy of your Version 5 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 Version 5, you risk losing some of your tables and data.

To create a copy of Version 5 catalog and directory, run Version 5 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 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 *DB2 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 6: 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 Version 5 in case you need to fall back from DB2 for OS/390 and z/OS Version 7.

## Making DB2 load modules available to TSO and batch users

If you included *prefix.SDSNEXIT* and *prefix.SDSNLOAD* in your LNKSTxx, you can skip this step.

If you have not included *prefix.SDSNEXIT* and *prefix.SDSNLOAD* in your LNKSTxx, you must add JOBLIB or STEPLIB statements to your logon procedures and JCL to ensure that you access the Version 7 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 227 for information on the items you can modify.

The DSNEMC01 CLIST uses the values specified on installation panel DSNTIPF and stores the results in the ISPF profile member DSNEPROF. Any customized DSNEPROF members can be migrated from Version 5 to Version 7. However, you need to examine any new or changed default panel values to ensure that your customized values are still valid.

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*.SDSNPFPP, *prefix*.SDSNPFPM, and *prefix*.SDSNPFPS. You must concatenate the DB2 English DB2I panels in *prefix*.SDSNPFPE or if you are using Kanji panels in *prefix*.SDSNPFPK to ISPPLIB. If you are using Online Help, include *prefix*.SDSNPFPT 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. Table 48 on page 248 shows the library table member names and their contents.

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(DSNPFPT) 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(DSNPFPT) F(DSNETBLS) REUSE
```

If you do not allocate the DSNPFPT 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 IKJ56861I.

For additional information on this TSO error message and how to *close* this data set, refer to *ISPF V4 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 371.

Refer to 290 for more information on using your TSO and CAF logon procedures.

---

## Migration step 7: 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 11. Connecting the IMS attachment facility” on page 425. This chapter also covers the requirements from a DB2 perspective and refers to IMS books for specific IMS information.

---

## Migration step 8: Connect CICS to DB2 (optional)

Connecting DB2 to CICS requires that you regenerate several CICS tables with additional entries. A macro is supplied with CICS to define the connection between CICS and DB2 using a resource control table (RCT).

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:

- 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 12. Connecting the CICS attachment facility” on page 433.

---

## Migration step 9: Stop DB2 Version 5 activity

Before making DB2 for OS/390 and z/OS Version 7 operational, ensure that all work is stopped on Version 5. If you do not stop work on Version 5, migration procedures might fail.

To stop work on Version 5, 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 -DSN1 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 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

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)

---

## Migration step 10: Back up your DB2 for OS/390 Version 5 volumes (optional)

At this point, you can back up your Version 5 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 11: Define DB2 initialization parameters: 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 Version 5 subsystem parameter module 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, DSNZPxxx 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 HDB7710, 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 Version 5 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.

**Recommendation:** Add 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 Part 4 (Volume 1) of *DB2 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.

For more information about access method services, see DFSMS/MVS: Access Method Services for VSAM Catalogs, SC26-4905-02.

---

## Migration step 12: 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 Part 3 (Volume 1) of *DB2 Administration Guide* for suggestions and instructions for including DB2 in your security system.

Because your Version 7 system reuses the data objects from your Version 5 system, you have probably already supplied the protection those objects need. However, you probably want to protect the new (Version 7) DB2 for OS/390 and z/OS data objects.

---

## Migration step 13: Define DB2 Version 7 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 Version 5 procedures so they do not conflict with Version 7 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.

**Recommendation:** For SYS1.PROCLIB, 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. 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 57.

## 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.

- LNKSTxx

Job DSNTIJMV updates this member to include the DB2 load module library, *prefix.SDSNLINK*, in the LNKSTxx. If you moved the modules from *prefix.SDSNLINK* into another library, edit DSNTIJMV to include that library in LNKSTxx. If you have combined *prefix.SDSNLINK* and *prefix.SDSNLOAD* into one library, edit DSNTIJMV to include the combined library in LNKSTxx. See *OS/390 MVS Initialization and Tuning Guide* for restrictions on data sets that are concatenated in LNKST.

**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 LNKSTxx) 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. DB2 can schedule synchronous read/write I/Os and prefetch read I/Os under the application address space's I/O scheduling priority. See Part 5 (Volume 2) of *DB2 Administration Guide* for more information about the effects of this type of I/O scheduling. To enable this type of I/O scheduling, 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 Version 5 procedures so they are not replaced by DB2 for OS/390 and z/OS Version 7 procedures.
  3. DSNTIJMV updates SYS1.PROCLIB to include the following Version 7 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 and z/OS Version 7 procedures replace your Version 5 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 5 procedures (such as data set names), make similar changes to the Version 7 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 48.

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 14: 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 new 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 15: Define user authorization exit routines: DSNTIJEX

Job DSNTIJEX builds the sample authorization exits, DSN3@SGN and DSN3@ATH and the user version of the access control authorization exit DSNX@XAC, 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 *DB2 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 *DB2 Administration Guide*. For more information on controlling data access, see Part 3 (Volume 1) of *DB2 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 or 4.

If job DSNTIJEX fails or abends, correct the problem and rerun the job.

DSNXSXAC is a copy of the default access control authorization exit that can be modified by the user. This exit allows you to bypass some or most of DB2 authorization checking and to specify your own authorization checking. If you do not modify it then this step is not needed and should be deleted.

---

## Migration step 16: 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 and z/OS Version 7. Be sure that the early code pre-conditioning PTFs have been installed on your system before you migrate. The Version 7 early code is *downward* compatible with Version 5.

Provided that you are at the appropriate service level for Version 5, 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 LNKSTxx 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 13: Define DB2 Version 7 to MVS: DSNTIJMV” on page 295 and “Choosing link list options” on page 57.

After you IPL MVS, message DSN3100I appears on the MVS console, stating that DB2 is ready for the START command.

---

## Migration step 17: Start DB2 for OS/390 and z/OS Version 7

Perform the following steps to start DB2 for OS/390 and z/OS Version 7:

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 DSNTIPO. If you did not specify a parameter module on panel DSNTIPO, 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 18: Tailor DB2 for OS/390 and z/OS Version 7 catalog: DSNTIJTC

DSNTIJTC invokes the CATMAINT utility to migrate your Version 5 catalog to the Version 7 catalog. Before running this job, ensure you have enough DASD space as described in “Disk requirements for the work file database” on page 29 or the job will fail. See Table 52 on page 275 for a list of new and changed indexes that might affect your work file database space needs. Complete all premigration activities listed under “Identify unsupported objects (DSNTIJPM)” on page 279 before running this job. CATMAINT will fail if any views on SYSCOLDIST(STATS) are found.

DSNTIJTC contains three steps. The first step of 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 7 objects. All IBM-supplied indexes are created or updated sequentially during the execution of DSNTIJTC. The second step of DSNTIJTC searches for unsupported objects. One message will be issued for each unsupported object that is found. The third step does the stored procedure migration processing.

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. All of these messages are written to the SYSPRINT data set. See the message descriptions in *DB2 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 at the correct maintenance level, correct the problem and run the job again. If you run the job again and the job still fails, return to Version 5. Because CATMAINT failures roll back all Version 7 changes, the catalog and directory are returned to their original premigration Version 5 format. Altered indexes are not rolled back. Determine if any index is in a pending status by using the CHECK INDEX utility. To return to Version 5 you need to:

- Rename procedures to use Version 5 libraries.
- Reconnect TSO, IMS, and CICS to Version 5 libraries.

See 304 for details on these steps.

If your Version 5 system is damaged you need to perform a point in time recovery. Follow these steps to recover your Version 5 system:

- Restore Version 5 catalog and directory from image copies.
- Restore BSDSs from archive logs taken prior to migration.
- Flush the SCA (for data sharing environments only).
- Recover the catalog and directory indexes.

For more details on recovering data to a prior point of consistency, see Part 4 (Volume 1) of *DB2 Administration Guide*.

---

## Migration step 19: Ensure there are no problems with the catalog (optional)

Check the integrity of your DB2 for OS/390 and z/OS Version 7 catalog and directory by following, in any order, these steps.

- Run CHECK INDEX on all the indexes in the catalog and directory using job DSNTIJCX.
- Run link checker (DSN1CHKR) to ensure that there are no existing broken links. See “Migration Step 2: Run link checker on DB2 for OS/390 Version 5 table spaces (optional)” on page 286 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 5 DSNTPE2 program to execute the queries.
- Run the DSN1COPY utility with the CHECK option on the catalog table spaces.

---

## Migration step 20: Prepare dynamic SQL program: DSNTIJTM

DSNTIJTM assembles, link-edits, binds, and runs DSNTIAD, a program that processes certain SQL statements dynamically. This job also assembles, link-edits, and binds DSNTWR, the load module for the DB2-supplied stored procedure WLM\_REFRESH, which is created in DSNTIJSG.

---

## Migration step 21: Bind SPUFI and DCLGEN and user maintained database activity: DSNTIJSG

In migration mode, job DSNTIJSG does the following:

- rebinds the IBM-defined packages
- alters the RLST
- creates and binds the stored procedure DSNWZP
- redefines the utility stored procedure
- redefines the DB2 SQL Procedures Processor stored procedure
- redefines the WLM refresh stored procedure

See Appendix B of *DB2 Utility Guide and Reference* for information about invoking utilities from a stored procedure. Stored procedures can be enabled after installation or migration. See “Enabling stored procedures after installation” on page 266 for the specific steps required to accomplish this.

If you chose a default value of DRDA for DATABASE PROTOCOL on panel DSNTIP5, and your DB2 Version 7 subsystem is in a data sharing group with DB2 subsystems that are at previous release levels, you need to bind the plans for DB2-supplied applications such as SPUFI and DCLGEN with the DBPROTOCOL(PRIVATE) option so that those applications are accessible to Version 5 members of the data sharing group. Edit the BIND commands for DB2-supplied applications in job DSNTIJSG.

If DSNTIJSG 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 7, DSNTIJSG rebinds SPUFI in Version 7. The Version 5 members cannot use a Version 7 SPUFI. If you attempt to run an SQL statement in a data sharing member at Version 5 with SPUFI at Version 7, 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 DSNTIJSG 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 and rerun DSNTIJSG. If you encounter resource shortages, review the parameters in job DSNTIJUZ, making any necessary modifications. Then stop DB2, rerun DSNTIJUZ, start DB2, and rerun DSNTIJSG from the last successful step.

---

## **Migration step 22: Migrate SQL procedures tables: DSNTIJMP (optional)**

If you used SQL procedures in Version 5 and you have data in the SYSIBM.SYSPSM and SYSIBM.SYSPSMOPTS tables, you must run DSNTIJMP. DSNTIJMP migrates data from SYSIBM.SYSPSM to catalog table SYSIBM.SYSROUTINES\_SRC and the data from SYSIBM.SYSPSMOPTS to SYSIBM.SYSROUTINES\_OPTS. This job also drops SYSIBM.SYSPSM and SYSIBM.SYSPSMOPTS and creates views of SYSIBM.SYSROUTINES\_SRC and SYSIBM.SYSROUTINES\_OPTS with the same column names as those of the old tables. The views are for use during coexistence or fallback. This step is optional only if you did not use SQL procedures in Version 5.

---

## **Migration step 23: Bind the packages for DB2 REXX Language Support: DSNTIJRX (optional)**

This step needs to be performed if you plan to use DB2 REXX Language Support. 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 24: Image copy DB2 for OS/390 and z/OS Version 7 catalog: DSNTIJIC**

Create a copy of the Version 7 DB2 for OS/390 and z/OS catalog and directory for backup purposes. See “Migration step 5: Image copy directory and catalog in case of fallback: DSNTIJIC” on page 288 for information on job DSNTIJIC.



---

## Migration step 25: Verify your DB2 for OS/390 and z/OS Version 7 system

Three steps verify your Version 7 subsystem. The first step, which is optional, runs selected Version 5 sample jobs. If the DB2 objects from the Version 5 sample jobs exist, you can follow the procedure under “Running the DB2 Version 5 sample jobs”. If you deleted the Version 5 sample objects continue with step 2 running the Version 7 sample jobs on your Version 7 subsystem. The final step is testing your application test cases.

### Running the DB2 Version 5 sample jobs

If all of the local DB2 objects from Version 5 still exist (that is, if you have not run job DSNTEJ0), run the selected Version 5 sample jobs listed below.

1. Change the JOBLIB statements to point to *prefix.SDSNLOAD*.
2. Be sure the DSN8EAE1 module you created when you originally ran the Version 5 sample jobs is copied to *prefix.SDSNEXIT*. DSN8EAE1 is an EDITPROC used by the employee sample table.
3. The Version 5 sample jobs must be edited before execution. Do NOT run all the Version 5 sample jobs. Only the specific jobs and job steps listed below should be run.
4. Test migration of the IVP phase 2 applications from Version 5:
  - a. DSNTEJ2A: Perform all but the first two steps of job DSNTEJ2A.  
Expect a return code of 4 because table spaces DSN8D51U.NEWDEPT and DSN8D51U.NEWPHONE are placed in copy pending states.
  - b. DSNTEJ2C: Execute only the RUN PROGRAM(DSN8BC3) PLAN(DSN8BH51) statement in step PH02CS04.
  - c. DSNTEJ2D: Execute only the RUN PROGRAM(DSN8BD3) PLAN(DSN8BD51) statement in step PH02DS03.
  - d. DSNTEJ2E: Execute only the RUN PROGRAM(DSN8BE3) PLAN(DSN8BE51) statement in step PH02ES04.
  - e. DSNTEJ2F: Execute only the RUN PROGRAM(DSN8BF3) PLAN(DSN8BF51) statement in step PH02FS03.
  - f. DSNTEJ2P: Run only step PH02PS05.
5. Test migration of the IVP phase 3 applications from Version 5:
  - a. Do not run job DSNTEJ3C or DSNTEJ3P.
  - b. If you want to test the DB2 Version 5's ISPF/CAF applications under Version 7, you must place the Version 5 SDSNSPFP panel library ahead of the Version 7 SDSNSPFP panel library in the ISPLIB concatenation. This is necessary in order for the plans migrated from Version 5 to be used. Be sure to remove the Version 5 SDSNSPFP library from your ISPLIB concatenation when you are finished testing the Version 5 IVP applications under Version 7.  
  
See the Version 5 *DB2 Installation Guide* for directions on how to start and run the Version 5 ISPF/CAF applications.

Do not run any other Version 5 sample jobs.

### Running the Version 7 sample jobs

The detailed instructions for running your Version 7 sample jobs on your Version 7 subsystem are described in “Chapter 10. Verifying with the sample applications” on page 349.

## Testing Version 7 with your application test cases

After completing the DB2 Version 7 sample jobs you should test your applications on the Version 7 subsystem. For information about testing application programs see *DB2 Application Programming and SQL Guide*.

---

## Falling back

Falling back is the process of returning to a previous version of DB2 after migrating your catalog and directory to DB2 for OS/390 and z/OS Version 7. You can fallback to Version 5 only after successfully migrating the catalog to Version 7 using job DSNTIJTC. Fall back if you have a severe error while operating Version 7 and you want to return to operation on the release from which you migrated. After fallback, the catalog remains a Version 7 catalog.

Remigrating is the process of returning to Version 7 after falling back to a previous version.

## Fallback considerations

To avoid complications, do not use the new DB2 for OS/390 and z/OS Version 7 facilities until you are certain that you will not need to fall back.

### Data sharing

There are additional considerations for falling back if any member of a data sharing group falls back. See *DB2 Data Sharing: Planning and Administration* for these details.

### Frozen objects

Falling back does not undo changes made to the catalog during migration to Version 7. The migrated catalog is used after fallback. Some objects in this catalog that have been affected by Version 7 function might become *frozen* objects after fallback. Frozen objects are unavailable, and they are marked with the release dependency marker I, J, or K. If an object is marked with a release dependency, it is never unmarked. The release dependency marker is listed in the IBMREQD column of catalog tables.

In general, objects that depend on the new facilities of DB2 for OS/390 and z/OS Version 7 are frozen after you fall back to Version 5 until you remigrate to Version 7. Table 53 on page 305 lists the objects that are frozen when falling back to Version 5. They are marked with the **release dependency markers I, J, or K**.

Table 53. Objects frozen when falling back to Version 5

RELEASE DEPENDENT MARK = I, J, or K	
	Plans that use any new syntax or objects <sup>(1)</sup>
	Packages that use any new syntax or objects <sup>(1)</sup>
	Views that use any new syntax or objects
	Tables and views with triggers
	Tables with check constraints that reference a cast function.
	8 KB and 16 KB table spaces and associated indexes <sup>(2)</sup>
	Plans or packages that reference user-defined functions
	Plans or packages that reference a Unicode object
	Plans or packages using any new bind options such as DBPROTOCOL(DRDA) <sup>(3)</sup>
	Plans or packages that have statements that use a default encoding scheme, and the default is Unicode.
	Stored procedures or user-defined functions with parameters that explicitly or implicitly use the Unicode encoding scheme.
	Distinct types for which the base type is a string with the Unicode encoding scheme.
	Trigger packages
	Queries with scrollable cursor syntax
	Tables with LOB columns
	LOB table spaces
	Auxiliary tables
	Indexes on auxiliary tables
	Tables with columns using the ROWID data type
	Tables defined with a user-defined distinct type
	Tables, table spaces, or databases that are defined with the Unicode encoding scheme.
	Tables defined with identity columns
	Table spaces that contain a table with an identity column.
	Any table space with a DSSIZE greater than 4GB and their associated partitioning indexes and nonpartitioning indexes with PIECESIZE greater than 4GB
<b>Notes:</b>	
1.	If a plan or package simply references a declared temporary table using the qualifier SESSION, but does not use a DELCARE GLOBAL TEMPORARY TABLE statement, the plan or package does not become frozen.
2.	The table spaces are marked with a release dependency. The indexes are not marked with a release dependency but are unavailable after fallback due to their association with the table spaces.
3.	The release dependency is marked whether the DBPROTOCOL(DRDA) option is specified on bind, rebind, or is the default defined in the subsystem parameter.

**Plans and packages** become frozen objects when they use new SQL syntax, new BIND options and attributes, or reference frozen objects. When plans and packages become frozen objects, the automatic rebind process is adversely affected. See “Automatic rebind” on page 306 for details.

### General-use Programming Interface

While operating in Version 5, you can determine if any of your objects are frozen by issuing the following SELECT statements:

```
SELECT NAME FROM SYSIBM.SYSPLAN
WHERE IBMREQD = 'I' OR IBMREQD = 'J' OR IBMREQD = 'K';
SELECT LOCATION, COLLID, NAME, VERSION FROM SYSIBM.SYSPACKAGE
WHERE IBMREQD = 'I' OR IBMREQD = 'J' OR IBMREQD = 'K';
SELECT CREATOR, NAME FROM SYSIBM.SYSVIEWS
WHERE IBMREQD = 'I' OR IBMREQD = 'J' OR IBMREQD = 'K';
SELECT CREATOR, NAME FROM SYSIBM.SYSINDEXES
WHERE IBMREQD = 'I' OR IBMREQD = 'J' OR IBMREQD = 'K';
SELECT CREATOR, NAME, TYPE FROM SYSIBM.SYSTABLES
```

```

WHERE IBMREQD = 'I' OR IBMREQD = 'J' OR IBMREQD = 'K';
SELECT DBNAME, NAME FROM SYSIBM.SYSTABLESPACE
WHERE IBMREQD = 'I' OR IBMREQD = 'J' OR IBMREQD = 'K';

```

## End of General-use Programming Interface

### Automatic rebind

After fallback, plans or packages bound in Version 7 will be automatically rebound on first execution in Version 5. See *DB2 Application Programming and SQL Guide* for more details on automatic rebind.

However, if after fallback, you try to use plans or packages that are frozen, the automatic rebind in Version 5 that takes place the first time you try to run the plan in Version 5 fails. To make the plans and packages that were not automatically rebound on Version 5 available, change the SQL statements or remove the reference to a frozen object, precompile the application programs, and explicitly BIND the plans and packages on Version 5.

The following SQL statements should be executed on the same release. Results will be unpredictable if you execute some statements on Version 7 and other statements on a previous release.

- CREATE TABLE PRIMARY KEY
- CREATE TABLE UNIQUE
- CREATE UNIQUE INDEX (to enforce primary key)
- CREATE UNIQUE INDEX (to enforce unique key)
- DROP INDEX (drop index enforcing primary key)
- DROP INDEX (drop index enforcing unique key)
- DROP INDEX (drop index enforcing referential constraint)
- CREATE TABLE FOREIGN KEY
- ALTER TABLE DROP PRIMARY KEY
- ALTER TABLE DROP UNIQUE
- ALTER TABLE DROP FOREIGN KEY

### Other fallback considerations

Before you fall back to Version 5, you must be aware of the following considerations:

#### **Key constraints:**

If DROP INDEX is run on Version 5, and the index is enforcing a unique key constraint, the unique key constraint will be dropped when the index is dropped. This is true even if the unique key was created on Version 7.

**ALTER partition:** You should remove all restrictive states from a table space before falling back. But, if fallback occurs while a partition of a table space is in REORG PENDING (REORP) state, the entire table space is unavailable. You must run REORG TABLESPACE (SHRLEVEL NONE) on the entire table space from the fallback release to remove the REORP status and rebalance the partitions. After you fall back to Version 5, you can not use the statement ALTER INDEX to modify the limit key boundaries and you cannot specify a range of partitions using the REORG utility.

**Changes to the MODIFY utility:** You should alter indexes with the COPY NO option before falling back to Version 5. In Version 5, you cannot run the MODIFY RECOVERY utility on table spaces with indexes that have the COPY YES option.

**Stored procedures:** If you created stored procedures with the Version 7 statement CREATE PROCEDURE, you will not be able to access those stored procedures if you need to fall back to Version 5. SQLCODE -204 is returned if you try to access a Version 7 stored procedure after fall back because the procedure name is not in catalog table SYSIBM.SYSPROCEDURES.

**No predictive governing:** You can use the Version 7 RLST after falling back, but you cannot take advantage of the predictive governing features. Rows with RLFFUNC = 6 or 7 are ignored.

**LOB considerations:** If any log records are found for LOBs, the applicable pages for the LOB table space are placed in the LPL.

**Optimization hints:** You can still use your Version 7 PLAN\_TABLE after falling back, but you cannot take advantage of the optimization hints that you have added to the PLAN\_TABLE.

**Utilities COPY, REPORT, and RECOVER:** You must use the Version 5 COPY and RECOVER utility jobs for backup and recovery after fallback. You can use utility REBUILD INDEX because of an alias. SYSIBM.SYSCOPY and SYSIBM.SYSLGRNX rows inserted for indexes in Version 7 are ignored by the backup and recovery utilities after fallback.

**UNLOAD utility:** An UNLOAD utility cannot be stopped or started from Version 5. You also cannot issue the TERM utility command to a stopped UNLOAD utility job from a release prior to Version 7. Stopped UNLOAD utilities should be terminated using the TERM utility command before fallback occurs; otherwise, SYSUTIL records for these UNLOAD utility jobs will remain in the system.

**More consistent DB2 restart times:** Falling back causes postponed abort units of recovery (URs) to be treated as inabort or in flight. Their backout is performed as part of the first restart in Version 5 and all pagesets are removed from their restart pending (RESTP) state or advisory restart pending (AREST) state. If there is outstanding backout work on the pageset on another data sharing member, the pending states are not removed. Therefore, you should try to resolve postponed abort URs with the command RECOVER POSTPONED before falling back.

A Version 5 system cannot request limited backout processing. However, if you enter a DISPLAY DATABASE command on Version 5, it is possible for that display to show the advisory restart pending (AREST) state of page sets that can result when Version 7 uses postponed abort processing. Similarly, if you issue a DISPLAY GROUP command on a Version 5 subsystem, a member that is both active and has indoubt URs is displayed as ACTIVE (in a Version 7 subsystem they are displayed as 'A I'). The 'A I' status means the object is active but has indoubt or postponed abort units of recovery outstanding. On fallback to Version 5, the 'A I' status will be changed to ACTIVE even if indoubts exist. However, on remigration to Version 7, the ACTIVE status will be changed back to 'A I' if the indoubts still exist.

**New buffer pool sizes:** A CREATE or ALTER DATABASE or TABLESPACE that specifies an 8KB or 16KB buffer pool name does not work after falling back. If

BUFFERPOOL is not specified on the CREATE TABLESPACE statement and the default buffer pool for the data base was an 8KB or 16KB buffer in Version 7, then SQLCODE -204 is returned.

**New buffer pool type:** If a buffer pool is defined as VPTYPE(DATASPACE) in Version 7, and you fallback to Version 5, this buffer pool becomes a primary type buffer pool and it is allocated with the specified VPSIZE unless the VPSIZE is larger than the allowable 1.6 GB in which case the default VPSIZE is used. The BSDS is not updated during fallback so that the old values can be used on remigration. On remigration to Version 7, the original VPTYPE and VPSIZE will be used. However, if you alter these attributes using ALTER BUFFERPOOL while in fallback mode on Version 5, the BSDS will be updated and the Version 7 attributes will be lost.

**New CLUSTERRATIOF column:** When you fallback to Version 5, the value in CLUSTERRATIOF will be used. If you run RUNSTATS on Version 5, the CLUSTERRATIOF column will be reset to the default value. Upon remigration to Version 7, the optimizer will use CLUSTERRATIO until RUNSTATS is run on Version 7.

**Changing column length in an index:** Changing the length of a column of an index in Version 7 may cause an error after falling back to Version 5. A resource unavailable message with RC '00C9009E' occurs if the APAR to enable immediate index access has not been applied.

**Changes to cascading REVOKE privileges:** When you fallback to Version 5 and attempt to revoke SYSADM or SYSCTRL privileges for a grantor and that revoke cascades down to Version 7 objects or new privilege grants on Version 5 objects, the revoke will fail with message DSNT501I.

**Changes to IMMEDIATE function:** If a Version 7 plan or package bound with IMMEDIATE(YES | PH1) is executed on a DB2 Version 5 member that does not have support for the IMMEDIATE (YES | PH1) attribute, the IMMEDIATE(YES | PH1) attribute is ignored at runtime. The Version 5 plan or package uses the default IMMEDIATE(NO) behavior without having to be rebound.

**Table and index spaces with undefined data sets:** You can use the DEFINE NO option on the CREATE TABLESPACE or CREATE INDEX statements to defer creating the VSAM data set until it is used. If the underlying VSAM data sets have not been created when you fall back to Version 5, you will receive an -904 SQLCODE for any SQL statements that refer to the table spaces and index spaces with undefined data sets. Do not use the REPAIR DBD or STOSPACE utilities on the undefined data sets.

## Fallback procedure

Because the structure of the DB2 for OS/390 and z/OS Version 7 catalog is used in Version 5 after falling back, the fallback procedure involves only a few steps:

1. Run Phase 0 of the Version 7 installation verification procedure
2. Stop Version 7 activity
3. Reactivate Version 5
4. Reconnect TSO, IMS, and CICS to Version 5
5. Start Version 5
6. Verify fallback

You can save your Version 7 TSO LOGON procedures and JCL for remigration to Version 7.

## Fallback Step 1: Run phase 0 of the DB2 for OS/390 and z/OS Version 7 installation verification procedure

This step removes all the installation verification processing done on DB2 for OS/390 and z/OS Version 7. When you run this step, all the DB2 for OS/390 and z/OS Version 7 sample tables and plans are deleted. You re-create these objects when you remigrate to Version 7.

## Fallback Step 2: Stop DB2 for OS/390 and z/OS Version 7 activity

Ensure that no recovery is required on system databases.

To stop Version 7 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 version of DSNTIJUZ was run. Check to see that you started DB2 with the correct subsystem parameter load module.

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.

### Fallback Step 3: Reactivate DB2 Version 5 code: DSNTIJFV

This job renames procedures to activate Version 5 and deactivate DB2 for OS/390 and z/OS Version 7. 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 5.

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.

### Fallback Step 4: Reconnect TSO, IMS, and CICS to DB2 Version 5

Reestablish your Version 5 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 and z/OS Version 7, reassemble the RCT with the Version 5 libraries.

If you did not overwrite the load module, change the STEPLIB statements for CICS and IMS jobs so they refer to the Version 5 libraries.

### Fallback Step 5: Start DB2 Version 5

Perform the following steps to start Version 5:

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 5 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 and z/OS Version 7 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 Version 5.
  - b. Determine which units of work are incomplete by scanning the DB2 recovery log with the DB2 for OS/390 and z/OS Version 7 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 7.
  - f. Start Version 5 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 and z/OS Version 7 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 Part 4 (Volume 1) of *DB2 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).

## Fallback 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 and z/OS Version 7 samples on Version 5.

### *How to verify fallback:*

1. Run the Version 5 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 5. These are described in “Other fallback considerations” on page 306.

---

## Remigrating

Migrating after falling back (remigrating) is simpler than the migration process described earlier in this chapter.

When remigrating, refer to “Migration Considerations” on page 271 because many of those considerations apply to remigrations, too. Which considerations apply depends on the type of activity that took place on your Version 5 subsystem after falling back.

A plan or package is automatically rebound in Version 7 when executed for the first time after remigration if it was not explicitly bound in Version 5. 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 7. This means the plan or package does not benefit from Version 7 enhancements.

When remigrating, you have to:

1. Run DSN1COPY with the CHECK option on the Version 5 catalog table spaces. Also, run DSN1CHKR on Version 5. For information on these utilities, see “Migration Step 2: Run link checker on DB2 for OS/390 Version 5 table spaces (optional)” on page 286. Finally, execute the queries in member DSNTESQ of *prefix.SDSNSAMP*.
2. Image copy the Version 5 catalog using the DSNTIJIC job.  
Although this step is not required, it is recommended. See “Migration step 5: Image copy directory and catalog in case of fallback: DSNTIJIC” on page 288 for more information.
3. Stop Version 5.
4. Reconnect TSO, IMS, and CICS to DB2 for OS/390 and z/OS Version 7.  
Reestablish your Version 7 logon procedures and JCL, as well as your Version 7 CICS and IMS connections.
5. Rebuild Version 7 cataloged procedures.

Rename the Version 7 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 7 to MVS, and run the job. (There is no need to define Version 7 to MVS a second time.)

6. Start DB2 for OS/390 and z/OS Version 7.

Make sure you are using your Version 7 subsystem parameter load module.

7. Image copy the Version 7 catalog using the DSNTIJIC job.

For information on job DSNTIJIC, see “Migration step 5: Image copy directory and catalog in case of fallback: DSNTIJIC” on page 288.

8. Verify your DB2 for OS/390 and z/OS Version 7 system.

For information on this procedure, see “Migration step 25: Verify your DB2 for OS/390 and z/OS Version 7 system” on page 303.



---

## Chapter 9. Migrating from Version 6 to Version 7

This chapter describes the steps necessary to migrate from DB2 Version 6.

Before you begin migration, you need to load the DB2 libraries. This process is described beginning on page 51. You also need to run the installation CLIST. The process is described beginning on page 77. Also see “Migration Considerations” for information on changes that might affect your migration process.

When you migrate to Version 7, avoid using new DB2 facilities until you are not likely to fall back.

**Attention:** If you do not follow documented procedures or if you try to migrate from a release other than Version 6, unpredictable results can occur after migration.

Make sure that your Version 6 subsystem is at the proper service level. The recommended maintenance level is the DB2 Version 6 refresh. You can obtain an equivalent service level by ordering a service update via PDO (level 0014) or ESO (level 0003) and applying service to PUT 0003. 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. Also check monthly for access to the most current information about DB2.

Before migrating the first member to Version 7, check the service levels of any coupling facilities running with coupling facility control code (CFCC) at CFLEVEL=7 or CFLEVEL=8. Coupling facilities at CFLEVEL=7 should have service level 1.06 or higher, and coupling facilities at CFLEVEL=8 should have service level 1.03 or higher. If these service levels are not installed, data corruption can occur. No service level requirements exist for the other CFLEVELs.

You can use the MVS D CF command to display the service level of coupling facilities. To use this command with releases of OS/390 previous to Version 2 Release 10, APAR OW38667 must be applied.

---

### Migration Considerations

Be aware of the following changes that might affect your migration to Version 7. For information on how migration might affect your DB2 operations, see “Make adjustments for release incompatibilities” on page 321.

#### Scrollable cursors are supported

Support for scrollable cursors enables random access to data in a table. Temporary result tables are stored in declared temporary tables. You must allow sufficient storage space for the new TEMP database and segmented table spaces.

#### Unicode support

If you plan to use Unicode, all members of a data-sharing group must convert to Version 7.

## **New default encoding scheme**

In Version 6, the default encoding scheme for host variables in the SET host variable and VALUES INTO statements was EBCDIC. In Version 7, the default encoding scheme is the value in the APPLICATION ENCODING SCHEME bind option.

## **Revoking SYSADM authorization**

If an authorization ID with SYSADM authority grants USAGE on a JAR in Version 7, another authorization ID cannot revoke its SYSADM authority in a previous release. You can revoke SYSADM authority only from Version 7.

## **Encoding schemes of string parameters for routines**

The new parameter CCSID clause allows you to define the encoding scheme of all string parameters for user-defined functions and stored procedures at once. In previous versions, you had to define a CCSID for each string parameter if you wanted an encoding scheme other than the default. Also, EBCDIC is no longer the default encoding scheme for system-defined parameters. DB2 now uses the same encoding scheme for both user-specified and system-generated string parameters.

## **Changed messages from the LOAD utility**

In previous releases of DB2, if the data types of input data and the column into which the data was loaded were incompatible, DB2 issued message DSNU390I during the UTILITY phase of utility processing and terminated the utility with return code 8. In Version 7, if the data types are incompatible, DB2 issues message DSNU299I during the RELOAD phase. If LOAD is not performing discard processing, LOAD abnormally terminates with reason code 00E40323. If LOAD is performing discard processing, LOAD issues message DSNU299I for every row of input data and puts the rejected records in the discard file. LOAD then terminates with return code 8.

## **Creating tables with DBCS and mixed columns**

You can no longer create extended binary-coded decimal interchange code (EBCDIC) tables with GRAPHIC, VARGRAPHIC, or DBCLOB columns when the setting for installation option MIXED DATA is NO. You also cannot alter EBCDIC tables to add GRAPHIC, VARGRAPHIC, or DBCLOB columns.

## **Increased size for generated code**

In DB2 Version 7, you can specify CCSIDs for host variables. Because DB2 stores the CCSIDs with the host variables, the generated source code and object code for a DB2 application might be larger than in previous versions.

## **Consider increasing IDBACK and CTHREAD**

Because utilities may use additional threads, you should consider increasing the IDBACK and CTHREAD subsystem parameters. Increasing these parameters can help you avoid failure of some utilities due to increased thread usage. An increase also supports the additional parallelism to be performed by the utilities.

## **Change in recording soft errors in SYS1.LOGREC data set**

In previous versions of DB2, abends that occur as a result of errors in SQL statements were handled by DB2 recovery routines and were recorded in SYS1.LOGREC. Examples of these errors include:

- An error in decimal arithmetic

- Overflow error
- Underflow error

In Version 7, by default, these abends are not recorded in SYS1.LOGREC. If you want these errors to be recorded in SYS1.LOGREC, you can specify YES in the RECORD SOFT ERRORS field of installation panel DSNTIPM.

## Customized DB2I defaults can be migrated

A DB2I TSO IPSF profile member from a prior release can be migrated to the current release. The DSNEMC01 CLIST uses the values specified on installation panel DSNTIPF and stores the results in the ISPF profile member DSNEPROF. Any customized DSNEPROF members can be migrated from Version 6 to Version 7. However, you need to examine any new or changed default panel values to ensure that your customized values are still valid.

## Migrating 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 3 of *DB2 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 Version 6 load library for each member in the data sharing group. For best availability, you can apply the SPE to one member at a time. You can have Version 6 systems with and without the SPE running at the same time. Stop and restart each member to pick up the change. Before migrating the first member of the data sharing group to Version 7, the fallback SPE must be applied to all members in the group.
4. Follow the procedure on migrating the data sharing group in Chapter 4 of *DB2 Data Sharing: Planning and Administration*. You must complete the migration of the first member of the data sharing group to Version 7 before starting any other members at Version 7.
5. To prepare for fallback, keep the subsystem parameter load module used by Version 6.

The CLIST edits different jobs for enabling data sharing and migrating a data sharing member. See Chapter 4 of *DB2 Data Sharing: Planning and Administration* for the list of jobs that are edited for each case.

## Work file database size calculations

The migration job DSNTIJTC creates and updates indexes on catalog tables. These indexes are created and updated *sequentially* during migration. The work file database is used for the sort of each index; DB2 needs enough work file storage to sort the largest of the indexes in Table 54. The migration fails if you do not have enough storage, so ensure that you have enough space before you begin. See “Disk requirements for the work file database” on page 29 for information about space requirements.

Table 54 shows the indexes that are new and changed for existing catalog tables.

Table 54. Indexes added or updated sequentially using the work file database

Catalog table name	Index name	Column names
SYSIBM.SYSTRIGGERS	SYSIBM.DSNOTX03	SCHEMA, TRIGNAME

Table 54. Indexes added or updated sequentially using the work file database (continued)

Catalog table name	Index name	Column names
SYSIBM.SYSROUTINES	SYSIBM.DSNOFX07	NAME, PARM_COUNT, ROUTINETYPE, SCHEMA, PARM_SIGNATURE, PARM1, PARM2, PARM3, PARM4, PARM5, PARM6, PARM7, PARM8, PARM9, PARM10, PARM11, PARM12, PARM13, PARM14, PARM15, PARM16, PARM17, PARM18, PARM19, PARM20, PARM21, PARM22, PARM23, PARM24, PARM25, PARM26, PARM27, PARM28, PARM29, PARM30
SYSIBM.SYSROUTINES	SYSIBM.DSNOFX08	JARSCHEMA, JAR_ID
SYSIBM.SYSVIEWS	SYSIBM.DSNVVX01	CREATOR, NAME, SEQNO, TYPE
SYSIBM.SYSVTREE	SYSIBM.DSNVTH01	CREATOR, NAME, TYPE

## SYSIBM.SYSPSM and SYSIBM.SYSPSMOPTS no longer used

Tables SYSIBM.SYSPSM and SYSIBM.SYSPSMOPTS are no longer used. Job DSNTIJMP migrates the data from these tables into new catalog tables. For more information, see “Migration step 22: Migrate SQL procedures tables: DSNTIJMP (optional)” on page 340.

## Release coexistence

This section highlights considerations for coexistence between a previous release and Version 7 in a data sharing and in a distributed environment.

### IRLM

As you apply IRLM service to members of a data sharing group, some members run with the newer service level and some with the older service level. Even though each member uses the same release level of IRLM, the different service levels can raise issues you must consider. For more information about IRLM coexistence, see *DB2 Data Sharing: Planning and Administration*.

### Data sharing

DB2 supports both Version 6 and Version 7 or both Version 5 and Version 7 members in a data sharing group, but not all three versions at once. DB2 will support the coexistence of only two releases at a time. To support two releases, you must first apply the fallback SPE to all Version 6 members of the group as described in “Migrating a data sharing group” on page 317. Release coexistence begins when you migrate the first data sharing member to Version 7. You must successfully migrate the first data sharing member to Version 7 before attempting to migrate the other data sharing members.

For the best availability, 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 7 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 Version 6. Secondly, you avoid the automatic rebind that occurs when any plan or package that is bound



on Version 7 is run on Version 6. It also avoids the automatic rebind that occurs when a Version 7-bound plan or package that was automatically rebound on Version 6 is later run on Version 7.

If it is not possible to enforce on which member a plan or package runs, consider how to handle binds and automatic rebinds while two releases are coexisting. For more information, see the description of the AUTOBIND option of installation panel DSNTIPO.

For detailed information about data sharing release coexistence considerations, see *DB2 Data Sharing: Planning and Administration*.

**TSO and CAF logon procedures:** 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. If you forget to update those procedures and jobs before migrating to any release subsequent to Version 7, those procedures and jobs will no longer work in that subsequent release.

For a detailed list of considerations for a data sharing group with multiple DB2 releases, see Chapter 4 of *DB2 Data Sharing: Planning and Administration*.

### Distributed environment

DB2 for OS/390 and z/OS 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 and z/OS can be used only when using DRDA access.

Other DRDA partners at DRDA level 4 can also take advantage of the functions introduced in this release of DB2 for OS/390 and z/OS.

---

## Migration step 1: Premigration activities

This step is included as a reminder to do the following tasks before migrating to Version 7:

- “Remove incomplete table definitions”
- “Save critical access paths (optional)” on page 320
- “DRDA support for three-part names (optional)” on page 321
- “Make adjustments for release incompatibilities” on page 321

### Remove incomplete table definitions

Before migrating to Version 7, you must remove all incomplete table definitions. The following query identifies existing incomplete table definitions that must be fixed:

```
SELECT  COLNO, TBCREATOR, TBNAME, REMARKS
FROM    SYSIBM.SYSCOLUMNS
WHERE   COLNO = 0;
```

There are several ways to resolve incomplete table definitions:

- Drop the table if it you do not need it.
- Complete the definition.

The REMARKS field for each table identified in the SELECT result lists the column numbers of unique key constraints that do not have enforcing indexes. Commas separate constraint column sets , and spaces separate column

numbers with constraints. Create a unique index for each constraint that is listed in the incomplete table definition. For example:

COLNO	TBCREATOR	TBNAME	REMARKS
0	SYSADM	TBTEST1	4 12 23,5 9 relname,11

There are three constraints that do not have enforcing indexes. One of the constraints is a referential constraint with a constraint name of "relname". The referential constraint name can be ignored because it is not needed for the purpose of completing the table definition. The following is an example of a SELECT statement to identify the column names corresponding to the column number in the REMARKS field:

```
SELECT  TBCREATOR, TBNAME, COLNO, NAME
FROM    SYSIBM.SYSCOLUMNS
WHERE   TBCREATOR = 'SYSADM'
AND     TBNAME = 'TBTEST1'
AND     COLNO IN ( 4, 12, 23, 5, 9, 11 );
```

Create indexes by using the column numbers from the SELECT results:

```
CREATE UNIQUE INDEX IXTEST1
ON SYSADM.TBTEST1( COL4, COL12, COL23 );
CREATE UNIQUE INDEX IXTEST2
ON SYSADM.TBTEST1( COL5, COL9 );
CREATE UNIQUE INDEX IXTEST3
ON SYSADM.TBTEST1( COL11 );
```

Creating the index may not remove the incomplete table definition. In this case, use the third method below to remove the incomplete table definitions.

- Proceed with migration and re-create the table. After migration, unload all data from the table, drop the table, and re-create the table with the constraints and indexes.

During migration to Version 7, DB2 will search SYSCOLUMNS for incomplete table definitions. If any are found, a warning message will be issued.

## Save critical access paths (optional)

Sometimes changes between releases of DB2 cause unwanted access path changes. Consult with your performance analysts to determine which queries are especially critical and ensure that a PLAN\_TABLE exists that contains the good access path.

EXPLAIN your queries before migrating. Because EXPLAIN requires a rebind, this could change your access paths. Therefore, extract the needed queries and then EXPLAIN them under a different application or program name. This protects the existing application while the access path information is obtained. Then, after the access paths for the extracted queries are validated, you can update the APPLNAME or PROGNAME columns of PLAN\_TABLE to the correct name. For more information about using access path hints, see Part 5 (Volume 2) of *DB2 Administration Guide*.

To make sure that you do not lose any rows in the current PLAN\_TABLE, use ALTER TABLE to add the new Version 7 columns to the existing PLAN\_TABLE. See Chapter 5 of *DB2 SQL Reference*.

## Examine all new and changed values for DB2I panels

The DB2I default panels DSNEOP01 and DSNEOP02, will not be initialized with the values specified during the installation CLIST process during a migration. The DB2I panel variables in the ISPF profile from the previous release will be used on the current release. Any customized DSNEPROF members will be migrated from Version 6 to Version 7. However, you must examine any new or changed default panel values to ensure that your customized values are still valid.

## DRDA support for three-part names (optional)

A new bind parameter, DBPROTOCOL, enables application programs that use three-part names for remote access to use DRDA protocol. Existing applications can be converted from private protocol to DRDA with minimal rework. Packages need to be rebound with the DBPROTOCOL parameter. The governing row in the RLST at the server needs to be changed from a row that governs by plan to one that governs by package.

## Make adjustments for release incompatibilities

These changes might affect your DB2 operations after migrating to Version 7.

### Adjust application programs

You might need to adjust your application programs because of the following release incompatibilities.

Rewrite the unsuccessful SQL statements by using correlated references, breaking up UNIONS or using OUTER JOINS.

**SQLCODE -538:** After migrating to DB2 Version 7, it is possible to receive SQLCODE -538 on an SQL CREATE TABLE FOREIGN KEY or SQL ALTER TABLE ADD FOREIGN KEY statement that previously executed successfully. If the foreign key references a parent key that has not been defined as a primary key or unique key in the parent table, the foreign key creation will fail. You must ensure that the parent key has been created as a primary key or unique key in the parent table.

**SQLCODE -669:** After migrating to DB2 Version 7, it is possible to receive SQLCODE -669 on an SQL DROP INDEX statement that previously executed successfully. If the dropped index is a unique index used to enforce a unique (primary or unique key) constraint or referential constraint, you must drop the constraint before you can drop the index.

It is possible to drop a unique index (for a unique key only) without first dropping the unique key constraint if the unique key was created in a release of DB2 prior to Version 7, and the unique key has no associated referential constraints.

**Adjust user-defined function calls for new built-in functions:** Several new built-in functions have been added such as DAYOFWEEK\_ISO, WEEK\_ISO, and IDENTITY\_VAL\_LOCAL. If you have user-defined functions, invoke them with a fully qualified name to avoid calling built-in functions that may have the same name. If the user-defined functions are not invoked with a fully qualified name and SYSIBM is first in the SQL path, the built-in function will be selected instead of the user-defined function.

**Changed default for CREATE FUNCTION statement:** CREATE FUNCTION statements executed in a previous release for table functions had FINAL CALL as the default. When these statements are executed in Version 7, NO FINAL CALL is the default.

**Changed encoding scheme inheritance:** Any CREATE TABLE, CREATE GLOBAL TEMPORARY TABLE, and DECLARE GLOBAL TEMPORARY TABLE statements with a LIKE clause, but no CCSID and IN clauses, will inherit the encoding scheme of the table being copied. Previously, these tables would have inherited the system default encoding scheme.

**Change applications that call DSNACCQC or DSNACCAV:** You may need to change applications that call DB2-supplied stored procedures DSNACCQC or DSNACCAV. The stored procedures are now defined to run in a WLM-established stored procedures address space. The single result set that is returned by DSNACCQC and the second result set that is returned by DSNACCAV contain one more column than in previous releases of DB2.

**SQL reserved words:** There are several new SQL reserved words in Version 7. Refer to *DB2 SQL Reference* for the list and adjust your applications accordingly.

**Changed return code for message DSNU185I:** The return code for DSNU185I has changed from return code 8 to return code 0. If you have applications that scan the return code of this message, you might need to modify them. In Version 6, utility processing stopped upon encountering an object created with the DEFINE NO attribute. These undefined objects were identified with message DSNU185I and return code 8. In Version 7, undefined objects are skipped and utility processing continues. Message DSNU185I is issued, but with return code 0.

**Changed default for identity columns:** In Version 7, the default for the START WITH value of an identity column when a value is not explicitly specified is the specified value for MINVALUE (for an ascending sequence) or MAXVALUE (for a descending sequence). In previous versions, there were no MINVALUE or MAXVALUE options, so if a value was not explicitly specified for START WITH when an identity column was specified, the default was 1.

**New reserved qualifier for tables, SESSION:** DB2 uses a new implicit qualifier, SESSION for a declared temporary table. All tables with a high level qualifier of SESSION will be treated as a declared temporary table. You must modify any existing tables with a high level qualifier of SESSION to use another qualifier.

Use the following query to identify existing tables, views, aliases, and synonyms that will be inaccessible because of the 'SESSION' qualifier:

---

**Product-sensitive Programming Interface**

```
SELECT 'TABLE', CREATOR, NAME
  FROM SYSIBM.SYSTABLES
 WHERE CREATOR = 'SESSION' AND TYPE IN ('T', 'G', 'X')
UNION
  SELECT 'VIEW', CREATOR, NAME
    FROM SYSIBM.SYSTABLES
   WHERE CREATOR = 'SESSION' AND TYPE = 'V'
UNION
  SELECT 'ALIAS', CREATOR, NAME
    FROM SYSIBM.SYSTABLES
   WHERE CREATOR = 'SESSION' AND TYPE = 'A'
UNION
  SELECT 'SYNONYM', CREATOR, NAME
    FROM SYSIBM.SYSSYNONYMS
   WHERE CREATOR = 'SESSION';
```

---

**End of Product-sensitive Programming Interface**

DB2 cannot know during the bind of the plan or package whether the static SQL statement that references a table name qualified by SESSION is for a persistent base table or for a declared temporary table. These static SQL statements will be incrementally bound at run-time when the static SQL statement is issued.

### **DCE security authentication no longer supported**

DCE support has been removed. Kerberos authentication has replaced DCE. For more information about creating Resource Access Control Facility (RACF) profiles, see Part 3 (Volume 1) of *DB2 Administration Guide*.

### **Fewer sort operations for queries**

A performance enhancement to DB2 results in fewer sort operations for queries that have ORDER BY clauses and have WHERE clauses with predicates of the form COL=constant. This enhancement can result in the following changes from previous releases of DB2:

- Changes in access paths, which can result in changes to EXPLAIN output or the elapsed time of a query.
- Some queries that previously received SQL code -136 (sort key length exceeds maximum) may now execute successfully.

### **Sysplex query parallelism in the PLAN\_TABLE**

If a parallel group being rebound to Version 7 is capable of sysplex query parallelism, and only one DB2 data sharing member is active, a single-CEC parallel plan is developed, and an **X** is placed in column PARALLEL\_MODE of table PLAN\_TABLE. If you do not want the **X**, bind the plan or package on a member that is installed with COORDINATOR=NO. The **X** for column PARALLEL\_MODE means that all available assisting DB2s are used at run time. You need to rebind all static plans that have a PARALLEL\_MODE of X after migrating to take advantage of this change.

### **DISPLAY BUFFERPOOL changes**

The DISPLAY BUFFERPOOL command syntax and output have changed. The new command provides more data sharing information required for understanding the level of inter-system sharing taking place. The DISPLAY BUFFERPOOL output no longer issues messages DSNB450I, DSNB451I, and DSNB452I. Several new messages with expanded information are generated by this command. For details on the changes see the DISPLAY BUFFERPOOL command in *DB2 Command Reference*.

### **Index changes**

Indexes that have been altered, rebuilt, reorged, or newly created can increase in size by one page per nonpartitioned index or one page per index partition. The leaf distribution column (LEAFDIST) value in SYSINDEXPART may increase for indexes that have been altered, rebuilt, reorged, or newly created.

### **Work space formulas changed for utilities**

Due to the larger table spaces and indexes available, the record header for several utility work data sets is lengthened by several bytes. The changed record headers which are used in several formulas for estimating work data set size affects the following utilities:

- CHECK DATA
- CHECK INDEX
- LOAD
- REBUILD INDEX
- REORG

The new calculations are explained in *DB2 Utility Guide and Reference* under the topic for defining work data sets for each utility.

### **Default for FASTSWITCH parameter is YES**

For some utilities, the FASTSWITCH parameter has a default of YES, which results in the creation of J0001 data sets. Because some tools and other utilities require data set names to match, you may experience incompatibilities. The fifth-level qualifier can vary from I0001 to J0001. For more information about renaming data sets, see Part 2 (Volume 1) of DB2 Administration Guide.

---

## **Migration Step 2: Run link checker on DB2 for OS/390 Version 6 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, 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 ignore this message, see description of DSN1985I in *DB2 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 Part 3 of *DB2 Utility Guide and Reference*. For more information on CHECK INDEX, see Part 2 of *DB2 Utility Guide and Reference*.

You should run the following query on your Version 6 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 Version 7 member DSNTESEQ of *prefix.SDSNSAMP*.

---

#### General-use Programming Interface

```
SELECT * FROM SYSIBM.SYSVOLUMES V1
      WHERE VOLID <math>\neq</math> '*' AND
      EXISTS (SELECT * FROM SYSIBM.SYSVOLUMES V2
              WHERE V1.SGNAME = V2.SGNAME AND
                    V2.VOLID = '*')
```

---

#### End of General-use Programming Interface

---

## Migration step 3: Determine which plans and packages are invalid after migration (optional)

Migrating to Version 7 renders some plans and packages invalid. Find out which ones are invalid by running the following queries on your Version 6 subsystem:

---

#### Product-sensitive Programming Interface

```
SELECT DISTINCT DNAME
      FROM SYSIBM.SYSPLANDEP
      WHERE BNAME IN ('DSNVVX01',
                     'DSNVTH01')
      AND BCREATOR = 'SYSIBM'
      AND BTYPE IN ('I','T')
      ORDER BY DNAME;
```

---

#### End of Product-sensitive Programming Interface

---

#### Product-sensitive Programming Interface

```
SELECT DISTINCT COLLID, NAME, VERSION
      FROM SYSIBM.SYSPACKDEP, SYSIBM.SYSPACKAGE
      WHERE BNAME IN ('DSNVVX01',
                     'DSNVTH01')
      AND LOCATION = ' '
      AND BQUALIFIER = 'SYSIBM'
      AND BTYPE IN ('I','T')
      AND COLLID = DCOLLID
      AND NAME = DNAME
      AND CONTOKEN = DCONTOKEN
      ORDER BY COLLID, NAME, VERSION;
```

---

#### End of Product-sensitive Programming Interface

These two queries are commented out in the Version 7 member DSNTESQ of *prefix.SDSNSAMP*. You can execute these queries from SPUFI or from a dynamic SQL program like DSNTDP2.

After migration, you can explicitly rebind these plans and packages, or let DB2 rebind them automatically. See Part 4 of *DB2 Application Programming and SQL Guide* for suggestions on rebinding these plans and packages.

---

## Migration step 4: 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 DSNTDP2.

Before you run these queries, you should have already run the DSN1CHKR utility and the CHECK INDEX utility in migration step 2.

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 5: Image copy directory and catalog in case of fallback: DSNTIJIC

### Attention:

You need to create a copy of your Version 6 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 Version 6, you risk losing some of your tables and data.

To create a copy of Version 6 catalog and directory, run Version 6 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 Version 6 job DSNTIJIC against the Version 6 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 *DB2 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 6: 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 Version 6 in case you need to fall back from DB2 for OS/390 and z/OS Version 7.

### Making DB2 load modules available to TSO and batch users

If you included *prefix*.SDSNEXIT and *prefix*.SDSNLOAD in your LNKSTxx, you can skip this step.

If you have not included *prefix*.SDSNEXIT and *prefix*.SDSNLOAD in your LNKSTxx, you must add JOBLIB or STEPLIB statements to your logon procedures and JCL to ensure that you access the Version 7 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 227 for information on the items you can modify.

The DSNEMC01 CLIST uses the values specified on installation panel DSNTIPF and stores the results in the ISPF profile member DSNEPROF. Any customized DSNEPROF members can be migrated from Version 6 to Version 7. However, you need to examine any new or changed default panel values to ensure that your customized values are still valid.

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 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, and *prefix*.SDSNSPFS. You must concatenate the DB2 English DB2I panels in *prefix*.SDSNSPFPE or if you are using Kanji panels in *prefix*.SDSNSPFPK to ISPPLIB. 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. Table 48 on page 248 shows the library table member names and their contents.

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 IKJ56861I.

For additional information on this TSO error message and how to *close* this data set, refer to *ISPF V4 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 371.

Refer to 327 for more information on using your TSO and CAF logon procedures.

---

## Migration step 7: 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 11. Connecting the IMS attachment facility” on page 425. This chapter also covers the requirements from a DB2 perspective and refers to IMS books for specific IMS information.

---

## Migration step 8: Connect CICS to DB2 (optional)

Connecting DB2 to CICS requires that you regenerate several CICS tables with additional entries. A macro is supplied with CICS to define the connection between CICS and DB2 using a resource control table (RCT).

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:

- 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 12. Connecting the CICS attachment facility” on page 433.

---

## Migration step 9: Stop DB2 Version 6 activity

Before making DB2 for OS/390 and z/OS Version 7 operational, ensure that all work is stopped on Version 6, including data sharing members if you have enabled data sharing. If you do not stop work on Version 6, fallback procedures may fail.

To stop work on Version 6, 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 10: Back up your DB2 for OS/390 Version 6 volumes (optional)

At this point, you can back up your Version 6 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 11: Define DB2 initialization parameters: 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 Version 6 subsystem parameter module 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, DSNZPxxx 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 HDB7710, 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 Version 6 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.

**Recommendation:** Add 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 Part 4 (Volume 1) of *DB2 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.

For more information about access method services, see DFSMS/MVS: Access Method Services for VSAM Catalogs, SC26-4905-02.

---

## Migration step 12: 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 Part 3 (Volume 1) of *DB2 Administration Guide* for suggestions and instructions for including DB2 in your security system.

Because your Version 7 system reuses the data objects from your Version 6 system, you have probably already supplied the protection those objects need. However, you probably want to protect the new (Version 7) DB2 for OS/390 and z/OS data objects.

---

## Migration step 13: Define DB2 Version 7 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 Version 6 procedures so they do not conflict with Version 7 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.

**Recommendation:** For SYS1.PROCLIB, 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. 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 57.

## 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.

- 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. DB2 can schedule synchronous read/write I/Os and prefetch read I/Os under the application address space's I/O scheduling priority. See Part 5 (Volume 2) of *DB2 Administration Guide* for more information about the effects of this type of I/O scheduling. To enable this type of I/O scheduling, 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 Version 6 procedures so they are not replaced by DB2 for OS/390 and z/OS Version 7 procedures.
  3. DSNTIJMV updates SYS1.PROCLIB to include the following Version 7 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 and z/OS Version 7 procedures replace your Version 6 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 6 procedures (such as data set names), make similar changes to the Version 7 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 48.

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 14: 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 new 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 15: Define user authorization exit routines: DSNTIJEX

Job DSNTIJEX builds the sample authorization exits, DSN3@SGN and DSN3@ATH and the user version of the access control authorization exit DSNX@XAC, 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 *DB2 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 *DB2 Administration Guide*. For more information on controlling data access, see Part 3 (Volume 1) of *DB2 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 or 4.

If job DSNTIJEX fails or abends, correct the problem and rerun the job.

DSNXSXAC is a copy of the default access control authorization exit that can be changed by the user. This exit allows you to bypass some or most of DB2 authorization checking and to specify your own authorization checking. If you do not change it then this step is not needed.

---

## Migration step 16: 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 and z/OS Version 7. Be sure that the early code pre-conditioning PTFs have been installed on your system before you migrate. The Version 7 early code is *downward* compatible with Version 6.

Provided that you are at the appropriate service level for Version 6, 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 LNKSTxx 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 13: Define DB2 Version 7 to MVS: DSNTIJMV” on page 295 and “Choosing link list options” on page 57.

After you IPL MVS, message DSN3100I appears on the MVS console, stating that DB2 is ready for the START command.

---

## Migration step 17: Start DB2 for OS/390 and z/OS Version 7

Perform the following steps to start DB2 for OS/390 and z/OS Version 7:

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 DSNTIPO. If you did not specify a parameter module on panel DSNTIPO, it defaults to DSNZPARM.

If DB2 starts successfully, 2 to 5 address spaces also start. These address spaces are *ssnm*MSTR and *ssnm*DBM1, and possibly *ssnm*DIST, *ssnm*SPAS, 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 18: Tailor DB2 for OS/390 and z/OS Version 7 catalog: DSNTIJTC

DSNTIJTC invokes the CATMAINT utility to migrate your Version 6 catalog to the Version 7 catalog. Before running this job, ensure you have enough DASD space as described in “Disk requirements for the work file database” on page 29 or the job will fail. See Table 52 on page 275 for a list of new and changed indexes that might affect your work file database space needs. Complete all premigration activities listed under “Identify unsupported objects (DSNTIJPM)” on page 279 before running this job.

#  
#  
#  
#  
#  
#

DSNTIJTC contains two steps for a migration from Version 6. The first step of 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 7 objects. All IBM-supplied indexes are created or updated sequentially during the execution of DSNTIJTC. The second step does the stored procedure migration processing.

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. All of

these messages are written to the SYSPRINT data set. See the message descriptions in *DB2 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 at the correct maintenance level, correct the problem and run the job again. If you run the job again and the job still fails, return to Version 6. Because CATMAINT failures roll back all Version 7 changes, the catalog and directory are in Version 6 format. Altered indexes are not rolled back. Determine if any index is in a pending status by using the CHECK INDEX utility. To return to Version 6 you need to:

- Rename procedures to use Version 6 libraries.
- Reconnect TSO, IMS, and CICS to Version 6 libraries.

See 341 for details on these steps.

If your Version 6 system is damaged you need to perform a point in time recovery. Follow these step to recover your Version 6 system:

- Restore Version 6 catalog and directory from image copies.
- Restore BSDSs from archive logs taken prior to migration.
- Flush the SCA (for data sharing environments only).
- Recover the catalog and directory indexes.

For more details on recovering data to a prior point of consistence, see Part 4 (Volume 1) of *DB2 Administration Guide*.

When you remigrate, run the job again. To execute DSNTIJTC, you must have installation SYSADM authority.

---

## Migration step 19: Ensure there are no problems with the catalog (optional)

Check the integrity of your DB2 for OS/390 and z/OS Version 7 catalog and directory by following, in any order, these steps.

- Run CHECK INDEX on all the indexes in the catalog and directory using job DSNTIJCX.
- Run link checker (DSN1CHKR) to ensure that there are no existing broken links. See “Migration Step 2: Run link checker on DB2 for OS/390 Version 6 table spaces (optional)” on page 324 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 6 DSNTPE2 program to execute the queries.
- Run the DSN1COPY utility with the CHECK option on the catalog table spaces.

---

## Migration step 20: Prepare dynamic SQL program: DSNTIJTM

DSNTIJTM assembles, link-edits, binds, and runs DSNTIAD, a program that processes certain SQL statements dynamically. This job also assembles, link-edits, and binds DSNTWR, the load module for the DB2-supplied stored procedure WLM\_REFRESH, which is created in DSNTIJSG.

---

## Migration step 21: Bind SPUFI and DCLGEN and user maintained database activity: DSNTIJSJG

In migration mode, job DSNTIJSJG does the following:

- rebinds the IBM-defined packages
- alters the RLST
- creates and binds the stored procedure DSNWZP
- redefines the utility stored procedure
- redefines the DB2 SQL Procedures Processor stored procedure
- redefines the DB2 WLM\_REFRESH stored procedure

See Appendix B of *DB2 Utility Guide and Reference* for information about invoking utilities from a stored procedure. Stored procedures can be enabled after installation or migration. See “Enabling stored procedures after installation” on page 266 for the specific steps required to accomplish this.

If you chose a default value of DRDA for DATABASE PROTOCOL on panel DSNTIP5, and your DB2 Version 7 subsystem is in a data sharing group with DB2 subsystems that are at previous release levels, you need to bind the plans for DB2-supplied applications such as SPUFI and DCLGEN with the DBPROTOCOL(PRIVATE) option so that those applications are accessible to non-Version 6 members of the data sharing group. Edit the BIND commands for DB2-supplied applications in job DSNTIJSJG.

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 7, DSNTIJSJG rebinds SPUFI in Version 7. The Version 6 members cannot use a Version 7 SPUFI. If you attempt to run an SQL statement in a data sharing member at Version 6 with SPUFI at Version 7, 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 and rerun DSNTIJSJG. If you encounter resource shortages, review the parameters in job DSNTIJSJG, making any necessary modifications. Then stop DB2, rerun DSNTIJSJG, start DB2, and rerun DSNTIJSJG from the last successful step.

---

## Migration step 22: Migrate SQL procedures tables: DSNTIJMP (optional)

If you used SQL procedures before DB2 Version 7 and you have data in the SYSIBM.SYSPSM and SYSIBM.SYSPSMOPTS tables, you must run DSNTIJMP. DSNTIJMP migrates data from SYSIBM.SYSPSM to catalog table SYSIBM.SYSROUTINES\_SRC and the data from SYSIBM.SYSPSMOPTS to SYSIBM.SYSROUTINES\_OPTS. This job also drops SYSIBM.SYSPSM and SYSIBM.SYSPSMOPTS and creates views of SYSIBM.SYSROUTINES\_SRC and SYSIBM.SYSROUTINES\_OPTS with the same column names as those of the old tables. The views are for use during coexistence or fallback. This step is optional only if you did not use SQL procedures in Version 6.

---

## Migration step 23: Bind the packages for DB2 REXX Language Support: DSNTIJRX (optional)

This step needs to be performed if you plan to use DB2 REXX Language Support. 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 24: Image copy DB2 for OS/390 and z/OS Version 7 catalog: DSNTIJIC

Create a copy of the Version 7 DB2 for OS/390 and z/OS catalog and directory for backup purposes. See “Migration step 5: Image copy directory and catalog in case of fallback: DSNTIJIC” on page 326 for information on job DSNTIJIC.

---

## Migration step 25: Verify your DB2 for OS/390 and z/OS Version 7 system

Three steps verify your Version 7 subsystem. The first step which is optional, runs selected Version 6 sample jobs. If the DB2 objects from the Version 6 sample jobs exist, you can follow the procedure under “Running the DB2 Version 5 sample jobs” on page 303. If you deleted the Version 6 sample objects, continue with step 2, running the Version 7 sample jobs on your Version 7 subsystem. The final step is testing your application test cases.

### Running the DB2 Version 6 sample jobs

If all of the local DB2 objects from Version 6 still exist (that is, if you have not run job DSNTIJO), run the selected Version 6 sample jobs listed below.

1. Change the JOBLIB statements to point to *prefix*.SDSNLOAD.
2. Be sure the DSN8EAE1 module you created when you originally ran the Version 6 sample jobs is copied to *prefix*.SDSNEXIT. DSN8EAE1 is an EDITPROC used by the employee sample table.

3. The Version 6 sample jobs must be edited before execution. Do NOT run all the Version 6 sample jobs. Only the specific jobs and job steps listed below should be run.
4. Test migration of the IVP phase 2 applications from Version 6:
  - a. DSNTEJ2A: Perform all but the first two steps of job DSNTEJ2A.  
Expect a return code of 4 because table spaces DSN8D61U.NEWDEPT and DSN8D61U.NEWPHONE are placed in copy pending states.
  - b. DSNTEJ2C: Execute only the RUN PROGRAM(DSN8BC3) PLAN(DSN8BH61) statement in step PH02CS04.
  - c. DSNTEJ2D: Execute only the RUN PROGRAM(DSN8BD3) PLAN(DSN8BD61) statement in step PH02DS03.
  - d. DSNTEJ2E: Execute only the RUN PROGRAM(DSN8BE3) PLAN(DSN8BE61) statement in step PH02ES04.
  - e. DSNTEJ2F: Execute only the RUN PROGRAM(DSN8BF3) PLAN(DSN8BF61) statement in step PH02FS03.
  - f. DSNTEJ2P: Run only step PH02PS05.
5. Test the migration of the IVP phase 3 applications from Version 6:
  - a. Do not run job DSNTEJ3C or DSNTEJ3P.
  - b. If you want to test the DB2 Version 6's ISPF/CAF applications under Version 7, you must place the Version 6 SDSNSPFP panel library ahead of the Version 7 SDSNSPFP panel library in the ISPLIB concatenation. This is necessary in order for the plans migrated from Version 6 to be used. Be sure to remove the Version 6 SDSNSPFP library from your ISPLIB concatenation when you are finished testing the Version 6 IVP applications under Version 7.  
  
See the Version 6 *DB2 Installation Guide* for directions on how to start and run the Version 6 ISPF/CAF applications.

Do not run any other Version 6 sample jobs.

## Running the Version 7 sample jobs

The detailed instructions for running your Version 7 sample jobs on your Version 7 subsystem are described in “Chapter 10. Verifying with the sample applications” on page 349.

## Testing Version 7 with your application test cases

After completing the DB2 Version 7 sample jobs you should test your applications on the Version 7 subsystem. For information about testing application programs see *DB2 Application Programming and SQL Guide*.

---

## Falling back

Falling back is the process of returning to a previous version of DB2 after migrating your catalog and directory to DB2 for OS/390 and z/OS Version 7. You can fallback to Version 6 only after successfully migrating the catalog to Version 7 using job DSNTIJTC. Fall back if you have a severe error while operating Version 7 and you want to return to operation on the release from which you migrated. After fallback, the catalog remains a Version 7 catalog.

Remigrating is the process of returning to Version 7 after falling back to a previous version.

## Fallback considerations

To avoid complications, do not use the new DB2 for OS/390 and z/OS Version 7 facilities until you are certain that you will not need to fall back.

### Data sharing

There are additional considerations for falling back if any member of a data sharing group falls back. See *DB2 Data Sharing: Planning and Administration* for these details.

### Frozen objects

Falling back does not undo changes made to the catalog during migration to Version 7. The migrated catalog is used after fallback. Some objects in this catalog that have been affected by Version 7 function might become *frozen* objects after fallback. Frozen objects are unavailable, and they are marked with the release dependency marker I or J. If an object is marked with a release dependency, it is never unmarked. The release dependency marker is listed in the IBMREQD column of catalog tables.

In general, objects that depend on the new facilities of DB2 for OS/390 and z/OS Version 7 are frozen after you fall back to Version 6 until you remigrate to Version 7. Table 55 lists the objects that are frozen when falling back to Version 6. They are marked with the **release dependency markers K**.

Table 55. Objects frozen when falling back to Version 6

RELEASE DEPENDENT MARK = K
Plans, packages, or views that use any new syntax or objects <sup>(1)</sup>
Queries with scrollable cursor syntax
Plans or packages using any new bind options such as DBPROTOCOL(DRDA) <sup>(2)</sup>
Plans or packages that reference a Unicode object
Plans or packages that have statements that use a default encoding scheme, and the default is Unicode.
Stored procedures or user-defined functions with parameters that explicitly or implicitly use the Unicode encoding scheme.
Distinct types for which the base type is a string with the Unicode encoding scheme.
Tables, table spaces, or databases that are defined with the Unicode encoding scheme.
User-defined functions and stored procedures written in Java™
User-defined functions written in SQL

#### Notes:

1. If a plan or package simply references a declared temporary table using the qualifier SESSION, but does not use a DECLARE GLOBAL TEMPORARY TABLE statement, the plan or package does not become frozen.
2. The release dependency is marked whether the DBPROTOCOL(DRDA) option is specified on bind, rebind, or is the default defined in the subsystem parameter.

**Plans and packages** become frozen objects when they use new SQL syntax, new BIND options and attributes, or reference frozen objects. When plans and packages become frozen objects, the automatic rebind process is adversely affected. See for details.

### General-use Programming Interface

While operating in Version 6, you can determine if any of your objects are frozen by issuing the following SELECT statements:



```

SELECT NAME FROM SYSIBM.SYSPLAN
WHERE IBMREQD = 'I' OR IBMREQD = 'K';
SELECT LOCATION, COLLID, NAME, VERSION FROM SYSIBM.SYSPACKAGE
WHERE IBMREQD = 'I' OR IBMREQD = 'K';
SELECT CREATOR, NAME FROM SYSIBM.SYSVIEWS
WHERE IBMREQD = 'I' OR IBMREQD = 'K';
SELECT CREATOR, NAME FROM SYSIBM.SYSINDEXES
WHERE IBMREQD = 'I' OR IBMREQD = 'K';
SELECT CREATOR, NAME, TYPE FROM SYSIBM.SYSTABLES
WHERE IBMREQD = 'I' OR IBMREQD = 'K';
SELECT DBNAME, NAME FROM SYSIBM.SYSTABLESPACE
WHERE IBMREQD = 'I' OR IBMREQD = 'K';

```

---

## End of General-use Programming Interface

### Automatic rebind

After fallback, plans or packages bound in Version 7 will be automatically rebound on first execution in Version 6. See *DB2 Application Programming and SQL Guide* for more details on automatic rebind.

However, if after fallback, you try to use plans or packages that are frozen, the automatic rebind in Version 6 that takes place the first time you try to run the plan in Version 6 fails. To make the plans and packages that were not automatically rebound on Version 6 available, change the offending SQL statements or remove the reference to a frozen object, precompile the application programs, and explicitly BIND the plans and packages on Version 6.

The following SQL statements should be executed on the same release. Results will be unpredictable if you execute some statements on Version 7 and other statements on a previous release.

- CREATE TABLE PRIMARY KEY
- CREATE TABLE UNIQUE
- CREATE UNIQUE INDEX (to enforce primary key)
- CREATE UNIQUE INDEX (to enforce unique key)
- DROP INDEX (drop index enforcing primary key)
- DROP INDEX (drop index enforcing unique key)
- DROP INDEX (drop index enforcing referential constraint)
- CREATE TABLE FOREIGN KEY
- ALTER TABLE DROP PRIMARY KEY
- ALTER TABLE DROP UNIQUE
- ALTER TABLE DROP FOREIGN KEY

### Other fallback considerations

Before you fall back to Version 6, you must be aware of the following considerations:

**Key constraints:** If DROP INDEX is run on Version 6, and the index is enforcing a unique key constraint, the unique key constraint will be dropped when the index is dropped. This is true even if the unique key was created on Version 7.

**Utilities COPY, REPORT, and RECOVER:** You must use the Version 6 COPY and RECOVER utility jobs for backup and recovery after fallback. You can use utility REBUILD INDEX because of an alias. SYSIBM.SYSCOPY and SYSIBM.SYSLGRNX rows inserted for indexes in Version 7 are ignored by the backup and recovery utilities after fallback.

**UNLOAD utility:** An UNLOAD utility cannot be stopped or started from Version 6. You also cannot issue the TERM utility command to a stopped UNLOAD utility job from a release prior to Version 7. Stopped UNLOAD utilities should be ended using the TERM utility command before fallback occurs; otherwise, SYSUTIL records for these UNLOAD utility jobs will remain in the system.

## Fallback procedure

Because the structure of the DB2 for OS/390 and z/OS Version 7 catalog is used in Version 6 after falling back, the fallback procedure involves only a few steps:

1. Run Phase 0 of the Version 7 installation verification procedure
2. Stop Version 7 activity
3. Reactivate Version 6
4. Reconnect TSO, IMS, and CICS to Version 6
5. Start Version 6
6. Verify fallback.

You can save your Version 7 TSO LOGON procedures and JCL for remigration to Version 7.

### Fallback Step 1: Run phase 0 of the DB2 for OS/390 and z/OS Version 7 installation verification procedure

This step removes all the installation verification processing done on DB2 for OS/390 and z/OS Version 7. When you run this step, all the DB2 for OS/390 and z/OS Version 7 sample tables and plans are deleted. You re-create these objects when you remigrate to Version 7.

### Fallback Step 2: Stop DB2 for OS/390 and z/OS Version 7 activity

Ensure that no recovery is required on system databases.

To stop Version 7 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 version of DSNTIJUZ was run. Check to see that you started DB2 with the correct subsystem parameter load module.

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.

### Fallback Step 3: Reactivate DB2 Version 6 code: DSNTIJFV

This job renames procedures to activate Version 6 and deactivate DB2 for OS/390 and z/OS Version 7. 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 6.

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.

### Fallback Step 4: Reconnect TSO, IMS, and CICS to DB2 Version 6

Reestablish your Version 6 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 and z/OS Version 7, reassemble the RCT with the Version 6 libraries.

If you did not overwrite the load module, change the STEPLIB statements for CICS and IMS jobs so they refer to the Version 6 libraries.

## Fallback Step 5: Start DB2 Version 6

Perform the following steps to start Version 6:

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 6 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 and z/OS Version 7 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 Version 6.
- b. Determine which units of work are incomplete by scanning the DB2 recovery log with the DB2 for OS/390 and z/OS Version 7 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 7.
  - f. Start Version 6 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 and z/OS Version 7 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 Part 4 (Volume 1) of *DB2 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).

## Fallback 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 and z/OS Version 7 samples on Version 6.

### **How to verify fallback:**

1. Run the Version 6 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 6. These are described in “Other fallback considerations” on page 343.

---

## Remigrating

Migrating after falling back (remigrating) is simpler than the migration process described in this chapter.

When remigrating, refer to “Migration Considerations” on page 315 because many of those considerations apply to remigrations, too. Which considerations apply depends on the type of activity that took place on your Version 6 subsystem after falling back.

A plan or package is automatically rebound in Version 7 when executed for the first time after remigration if it was not explicitly bound in Version 6. 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 7. This means the plan or package does not benefit from Version 7 enhancements.

When remigrating, you have to:

1. Run DSN1COPY with the CHECK option on the Version 6 catalog table spaces. Also, run DSN1CHKR on Version 6. For information on these utilities, see “Migration Step 2: Run link checker on DB2 for OS/390 Version 6 table spaces (optional)” on page 324. Finally, execute the queries in member DSNTESQ of *prefix.SDSNSAMP*.
2. Image copy the Version 6 catalog using the DSNTIJIC job.  
Although this step is not required, it is recommended. See “Migration step 5: Image copy directory and catalog in case of fallback: DSNTIJIC” on page 326 for more information.
3. Stop Version 6.
4. Reconnect TSO, IMS, and CICS to DB2 for OS/390 and z/OS Version 7.  
Reestablish your Version 7 logon procedures and JCL, as well as your Version 7 CICS and IMS connections.
5. Rebuild Version 7 cataloged procedures.  
Rename the Version 7 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 7 to MVS, and run the job. (There is no need to define Version 7 to MVS a second time.)
6. Start DB2 for OS/390 and z/OS Version 7.  
Make sure that you are using your Version 7 subsystem parameter load module.
7. Image copy the Version 7 catalog using the DSNTIJIC job.  
For information on job DSNTIJIC, see “Migration step 5: Image copy directory and catalog in case of fallback: DSNTIJIC” on page 326.
8. Verify your DB2 for OS/390 and z/OS Version 7 system.  
For information on this procedure, see “Migration step 25: Verify your DB2 for OS/390 and z/OS Version 7 system” on page 340.

---

## Chapter 10. 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**, the recommendation is that you run portions of the sample applications from the release you migrated from after you finish migration. This verifies the migration and ensures that the old jobs work with DB2 for OS/390 and z/OS Version 7. Version 7 sample applications can be run next. For information on how to run the sample applications from the previous release, see “Migration step 25: Verify your DB2 for OS/390 and z/OS Version 7 system” on page 303 or “Migration step 25: Verify your DB2 for OS/390 and z/OS Version 7 system” on page 340.

The installation verification procedure consists of eight phases: seven verification phases and one cleanup phase that drops sample objects. Each of the seven 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 7 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 6.

These jobs have been tailored and loaded by the installation 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 *DB2 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 7 and to fall back to Version 6 in case of a Version 7 failure. If you are migrating, you can run the fallback release sample programs on Version 7 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.

The JCL provided for CICS and IMS sets up transaction identifiers for the sample applications.

---

## Installation verification phases and programs

Table 56 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 DSNTSJ2C, you must use COB2 for all COBOL verification programs.

Run Phase 0 (job DSNTSJ0) 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, including user-defined functions. Phase 4 is for IMS users only, and Phase 5 is for CICS users only. Phase 6 sets up the sample tables and stored procedures for distributed processing. Phase 7 tests the LOB feature with sample tables, data and programs.

*prefix*.SDSNSAMP contains the program source. Details on how to print it are provided in “Printing the sample application listings” on page 396.

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 56 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 227.

After running the verification jobs, you can still fall back to Version 6. See “Falling back” on page 304 or “Falling back” on page 341 for details.

*Table 56. Relationship of phases to programs*

| Phase | Job                   | Program  | Description                                                 |
|-------|-----------------------|----------|-------------------------------------------------------------|
| 0     | DSNTSJ0               | DSNTIAD  | Remove sample applications and sample schema authorizations |
| 1     | DSNTSJ1               | DSNTIAD  | Create tables                                               |
|       |                       | DSN8CA   | Assembler interface to call attach facility                 |
|       |                       | DSN8EAE1 | Edit exit                                                   |
|       |                       | DSN8HUFF | Huffman compression exit                                    |
|       |                       | DSNUTILB | Utilities                                                   |
|       | DSNTSJ1L              | DSNTEP2  | Dynamic SQL program                                         |
|       | DSNTSJ1P              | DSNTEP2  | Dynamic SQL program                                         |
|       | DSNTSJ1S <sup>1</sup> | DSNHSP   |                                                             |
|       | DSNTSJ1T <sup>2</sup> | DSNUPROC |                                                             |
|       | DSNTSJ1U              | DSNTIAD  | Create Unicode table                                        |
|       |                       | DSNUTILB | Load Unicode table                                          |
|       |                       | DSNTEP2  | Select Unicode table                                        |
| 2     | DSNTSJ2A              | DSNTIAD  | Grant execution                                             |
|       |                       | DSNTIAUL | Unload and load tables                                      |
|       |                       | DSNUTILB | Utilities                                                   |
|       | DSNTSJ2C              | DCLGEN   | Generate declarations                                       |
|       |                       | DSNTIAD  | Grant execution                                             |



Table 56. Relationship of phases to programs (continued)

| Phase | Job                   | Program  | Description                                                          |
|-------|-----------------------|----------|----------------------------------------------------------------------|
|       |                       | DSN8BC3  | COBOL phone application                                              |
|       | DSNTEJ2D              | DSNTIAD  | Grant execution                                                      |
|       |                       | DSN8BD3  | C phone application                                                  |
|       | DSNTEJ2E              | DSN8MDG  | Prepare error message routine                                        |
|       |                       | DSN8BECL | Prepare classes used by C++ phone application                        |
|       |                       | DSNTIAD  | Grant execution                                                      |
|       |                       | DSN8BE3  | C++ phone application                                                |
|       | DSNTEJ2F              | DSNTIAD  | Grant execution                                                      |
|       |                       | DSN8BF3  | Fortran phone application                                            |
|       | DSNTEJ2P              | DCLGEN   | Generate declarations                                                |
|       |                       | DSNTIAD  | Grant execution                                                      |
|       |                       | DSN8BP3  | PL/I phone application                                               |
|       | DSNTEJ2U <sup>3</sup> | DSNTIAD  | Register sample user-defined functions                               |
|       |                       | DSN8DUAD | C program for ALTDAT function (current date)                         |
|       |                       | DSN8DUCD | C program for ALTDAT function (given date)                           |
|       |                       | DSN8DUAT | C program for ALTIME function (current time).                        |
|       |                       | DSN8DUCT | C program for ALTIME function (given time)                           |
|       |                       | DSN8DUCY | C program for CURRENCY function                                      |
|       |                       | DSN8DUTI | C program for TABLE_NAME, TABLE_SCHEMA, and TABLE_LOCATION functions |
|       |                       | DSN8EUDN | C++ program for DAYNAME function                                     |
|       |                       | DSN8EUMN | C++ program for MONTHNAME function                                   |
|       |                       | DSNTEP2  | Dynamic SQL program                                                  |
|       |                       | DSN8DUWF | User defined table function sample                                   |
|       |                       | DSN8DUWC | C program on client for user defined table function sample           |
| 3     |                       | SPUFI    | Sample SPUFI input                                                   |
|       | DSNTEJ3C              | DSNTIAD  | Grant execution                                                      |
|       |                       | DSN8CC   | COBOL interface to call attachment facility                          |
|       |                       | DSN8SCM  | COBOL connection manager                                             |
|       |                       | DSN8SC3  | COBOL phone application                                              |
|       |                       | DSN8HC3  | COBOL organization application                                       |
|       | DSNTEJ3P              | DSNTEP2  | Dynamic SQL application                                              |
|       |                       | DSNTIAD  | Grant execution                                                      |

Table 56. Relationship of phases to programs (continued)

| Phase | Job                   | Program | Description                                                                                        |
|-------|-----------------------|---------|----------------------------------------------------------------------------------------------------|
|       |                       | DSN8SPM | PL/I connection manager                                                                            |
|       |                       | DSN8SP3 | PL/I phone application                                                                             |
| 4     | DSNTEJ4C              | DSNTIAD | Grant execution                                                                                    |
|       |                       | DSN8ICx | Organization application                                                                           |
|       |                       | DSN8MCx | Copy code                                                                                          |
|       | DSNTEJ4P              | DSNTIAD | Grant execution                                                                                    |
|       |                       | DSN8IPx | Organization, project applications                                                                 |
|       |                       | DSN8MPx | Copy code                                                                                          |
| 5     | DSNTEJ5A              | DSNTIAC | CICS SQLCA formatter front-end                                                                     |
|       | DSNTEJ5C              | DSNTIAD | Grant execution                                                                                    |
|       |                       | DSN8CCx | Organization application                                                                           |
|       |                       | DSN8MCx | Copy code                                                                                          |
|       | DSNTEJ5P              | 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 |
|       | DSNTEJ6D <sup>4</sup> | DSN8ED1 | Compile, link-edit, bind, and run stored procedure sample application                              |
|       | DSNTEJ6T <sup>4</sup> | DSN8ED2 | Register, prepare, and bind the stored procedure sample application                                |
|       | DSNTEJ6P <sup>5</sup> | DSN8EP1 | Invoke the sample stored procedure                                                                 |
|       | DSNTEJ6S <sup>5</sup> | DSN8EP2 | Create sample stored procedure                                                                     |
|       | DSNTEJ6U <sup>6</sup> | DSN8EPU | Compile, link-edit, bind, and run stored procedure to execute a utility                            |
|       | DSNTEJ6V <sup>7</sup> | DSN8EE0 | C++ class that calls DSNTIAR                                                                       |
|       |                       | DSN8EE1 | C++ class that calls DSNUTILS                                                                      |
|       |                       | DSN8EE2 | C++ client for DSN8EE1                                                                             |
|       | DSNTEJ6W <sup>7</sup> | DSN8ED6 | Calls the WLM_REFRESH stored procedure                                                             |
|       | DSNTEJ6Z <sup>7</sup> | DSN8ED7 | C class that calls DSNWZP stored procedure                                                         |
|       | DSNTEJ61 <sup>8</sup> | DSN8EC1 | Create sample ODBA stored procedure                                                                |
|       | DSNTEJ62 <sup>8</sup> | DSN8EC2 | Invoke sample ODBA stored procedure                                                                |
|       | DSNTEJ63 <sup>9</sup> | DSN8ES1 | Run sample SQL procedure that calculates employee earnings.                                        |
|       | DSNTEJ64 <sup>9</sup> | DSN8ED3 | Call the sample SQL procedure, DSN8ES1 from a client                                               |
|       | DSNTEJ65 <sup>9</sup> | DSN8ED4 | Call the SQL procedure processor, DSNTPSMP                                                         |

Table 56. Relationship of phases to programs (continued)

| Phase | Job                    | Program  | Description                                        |
|-------|------------------------|----------|----------------------------------------------------|
| 7     | DSNTEJ7                | DSN8ES2  | Run SQL procedure that calculates employee bonuses |
|       |                        | DSN8ED5  | Call SQL procedure, DSN8ES2                        |
|       |                        | DSNTIAD  | Create sample LOB table                            |
|       |                        | DSNTIAD  | Create synonyms, grant access to LOB tables        |
|       |                        | DSNUTILB | Load sample LOB table                              |
|       | DSNTEJ71               | DSNUTILB | Produce statistics for LOB table spaces            |
|       |                        | DSNTIAD  | Grant access to plans                              |
|       |                        | DSN8DLPL | Populate sample LOB table with BLOB data           |
|       |                        | DSN8DLTC | Verify contents of LOB table                       |
|       |                        | DSNTIAD  | Grant access to plans                              |
| 10    | DSNTEJ75 <sup>10</sup> | DSN8DLRV | C employee resume application                      |
|       |                        | DSNTIAD  | Grant access to plans                              |
|       |                        | DSN8DLPV | C employee photo application                       |

Table 56. Relationship of phases to programs (continued)

| Phase        | Job | Program                                                                                                                                                                                                                                                                                                                                                           | Description |
|--------------|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| <b>Note:</b> |     |                                                                                                                                                                                                                                                                                                                                                                   |             |
|              | 1.  | Job DSNTEJ1S, 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 Part 2 (Volume 1) of <i>DB2 Administration Guide</i>                                                                                                                  |             |
|              | 2.  | Job DSNTEJ1T, used for adding rows to SYSIBM.SYSSTRINGS for character conversion purposes, is not a part of the sample applications to verify installation.                                                                                                                                                                                                       |             |
|              | 3.  | Job DSNTEJ2U is not created unless you specify C/C++ for OS/390 Version 1 Release 3 or subsequent releases at installation time.                                                                                                                                                                                                                                  |             |
|              | 4.  | Jobs DSNTEJ6T and DSNTEJ6D are not edited by the CLIST unless a REMOTE LOCATION name is entered on panel DSNTIPY, a LOCATION NAME is entered on panel DSNTIPR, a non-blank value is specified for DB2 PROC NAME on panel DSNTIPX, and you specify AUTO or COMMAND for DDF STARTUP OPTION on panel DSNTIPR.                                                        |             |
|              | 5.  | Jobs DSNTEJ6P and DSNTEJ6S are not edited by the CLIST unless a REMOTE LOCATION name is entered on panel DSNTIPY, a LOCATION NAME is entered on panel DSNTIPR, a non-blank value is specified for DB2 PROC NAME on panel DSNTIPX, you specify AUTO or COMMAND for DDF STARTUP OPTION on panel DSNTIPR, and you specify PL/I for MVS and VM on panel DSNTIPG.      |             |
|              | 6.  | Job DSNTEJ6U is not edited by the CLIST unless you specify PL/I for MVS and VM on panel DSNTIPG.                                                                                                                                                                                                                                                                  |             |
|              | 7.  | Job DSNTEJ6V is not edited by the CLIST unless you specify AUTO or COMMAND for the DDF STARTUP option on panel DSNTIPR and provide a value for WLM ENVIRONMENT on DSNTIPX.                                                                                                                                                                                        |             |
|              | 8.  | Jobs DSNTEJ61 and DSNTEJ62 are not edited by the CLIST unless a REMOTE LOCATION name is entered on panel DSNTIPY, a LOCATION NAME is entered on panel DSNTIPR, a non-blank value is specified for WLM PROC NAME on panel DSNTIPX, you specify AUTO or COMMAND for DDF STARTUP OPTION on panel DSNTIPR, and you specify IBM COBOL for MVS and VM on panel DSNTIPQ. |             |
|              | 9.  | Jobs DSNTEJ63, DSNTEJ64, and DSNTEJ65 are not edited by the CLIST unless a LOCATION NAME is entered on panel DSNTIPR, a non-blank value is specified for DB2 PROC NAME on panel DSNTIPX, and you specify AUTO or COMMAND for DDF STARTUP OPTION on panel DSNTIPR.                                                                                                 |             |
|              | 10. | Job DSNTEJ75 is not edited by the CLIST unless the GDDM MACLIB and GDDM LOAD MODULES fields on panel DSNTIPW are non-blank.                                                                                                                                                                                                                                       |             |

## 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 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 57 or the COBOL options in Table 59 on page 356 or the COB2 options in Table 58 on page 356. 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 57. 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 <sup>3</sup>  |
| NOCOMPILE(S)  | NOOFFSET                                 | RMODE(AUTO)        |
| NOCURRENCY    | NOOPTIMIZE                               | SIZE(MAX)          |
| NODBCS        | NOSEQUENCE                               | SOURCE             |
| NODECK        | NOSSRANGE                                | SPACE(1)           |
| NODUMP        | NOTERM <sup>1</sup> or TERM <sup>2</sup> | TRUNC(STD)         |
| NODYNAM       | NOTEST                                   | ZWB                |

**Notes:**

<sup>1</sup> Refers to jobs DSNTJ2C, DSNTJ3C, and DSNTJ4C only

<sup>2</sup> Refers to job DSNTJ5C only

<sup>3</sup> See the CICS documentation for actual options to use

Table 58. COB2 (vs COBOL II) options

|                                        |                                              |
|----------------------------------------|----------------------------------------------|
| ADV                                    | NOMAP                                        |
| APOST                                  | NONAME                                       |
| BUFSIZE(4096)                          | NONUMBER                                     |
| DATA(31)                               | NOOFFSET                                     |
| FLAG(I)                                | NOOPTIMIZE                                   |
| LANGUAGE(EN)                           | NOSOURCE <sup>1</sup> or SOURCE <sup>2</sup> |
| LIB <sup>1</sup> or NOLIB <sup>2</sup> | NOSSRANGE                                    |
| LINECOUNT(60)                          | NOTERM                                       |
| NOAWO                                  | NOTEST                                       |
| NOCMPR2                                | NOVBREF                                      |
| NOCOMPILE(S)                           | NOWORD                                       |
| NODBCS                                 | NOXREF <sup>1</sup> or XREF <sup>2</sup>     |
| NODECK                                 | NUMPROC(NOPFD)                               |
| NODUMP                                 | OBJECT                                       |
| NODYNAM                                | OUTDD(SYSOUT)                                |
| NOEXIT                                 | RENT <sup>3</sup>                            |
| NOFASTSRT                              | RESIDENT <sup>3</sup>                        |
| NOFDUMP                                | SEQUENCE                                     |
| NOFLAGMIG                              | SIZE(MAX)                                    |
| NOFLAGSA                               | SPACE(1)                                     |
| NOFLAGSTD                              | TRUNC(OPT)                                   |
| NOLIST                                 | ZWB                                          |

**Notes:**

- <sup>1</sup> Refers to job DSNTJ5C only
- <sup>2</sup> Refers to jobs DSNTJ2C, DSNTJ3C, and DSNTJ4C only
- <sup>3</sup> See the CICS documentation for actual options to use

Table 59. COBOL options

|                        |                         |                          |
|------------------------|-------------------------|--------------------------|
| ADV                    | NODECK                  | NOSYMDMP                 |
| APOST                  | NODMAP                  | NOSYNTAX                 |
| BUF=12288 <sup>1</sup> | NODYNAM                 | NOTERM                   |
| BUF=64000 <sup>2</sup> | NOENDJOB                | NOTEST                   |
| COMPILE=01             | NOFDECK                 | NOTRUNC                  |
| DUMP                   | NOFLOW                  | NOVBREF                  |
| FLAGW                  | NOLIB                   | NOVBSUM                  |
| LANGVL(2)              | NOLST <sup>1</sup>      | SEQ                      |
| LCOL2                  | NOLVL                   | SOURCE <sup>1</sup>      |
| LINECNT=57             | NOMIGR                  | or NOSOURCE <sup>2</sup> |
| LOAD                   | NONAME                  | SPACE1                   |
| LSTCOMP <sup>2</sup>   | NONUM                   | SXREF <sup>3</sup>       |
| L120 <sup>1</sup>      | NOOPTIMIZE              | or NOSXREF <sup>4</sup>  |
| L132 <sup>2</sup>      | NOPMAP                  | SYST                     |
| NOBATCH                | NOPRINT                 | VERB                     |
| NOCDECK                | NORESIDENT <sup>7</sup> | XREF <sup>5</sup>        |
| NOCLIST                | NOSTATE                 | or NOXREF <sup>6</sup>   |
| NOCOUNT                | NOSUPMAP                | ZWB                      |

**Notes:**

- <sup>1</sup> Refers to jobs DSNTJ2C, DSNTJ3C, and DSNTJ4C only
- <sup>2</sup> Refers to job DSNTJ5C only
- <sup>3</sup> Refers to jobs DSNTJ2C and DSNTJ4C only
- <sup>4</sup> Refers to jobs DSNTJ3C and DSNTJ5C only
- <sup>5</sup> Refers to job DSNTJ3C only
- <sup>6</sup> Refers to jobs DSNTJ2C, DSNTJ4C, and DSNTJ5C only
- <sup>7</sup> 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 the recommendation is that you use VS COBOL II or IBM COBOL for MVS & VM).

If you wish to run your sample 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.SDSNSAMP* and *prefix.NEW.SDSNTEMP* 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.SDSNSAMP* 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 *DB2 Command Reference*.

For more detailed instructions, see *IBM COBOL for MVS & VM Programming Guide* and *OS/390 Language Environment for OS/390 & 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 60 or Table 61 on page 358.

Table 60. C language options

|                         |                         |                           |                          |                         |
|-------------------------|-------------------------|---------------------------|--------------------------|-------------------------|
| NOAGGR                  | NOEVENTS                | NOHWOPTS                  | NOMEMORY                 | NOPPONLY                |
| NOALIAS                 | EXECOPS                 | NOINLINE <sup>4</sup>     | NESTINC(255)             | REDIR <sup>1</sup>      |
| ARGPARSE <sup>1</sup>   | NOEXPMAC                | LIST                      | OBJECT                   | NORENT                  |
| NOCHECKOUT <sup>2</sup> | NOEXPORTAL <sup>3</sup> | NOLOCALE                  | NOOE <sup>1</sup>        | NOSEARCH                |
| NOCSECT                 | FLAG(I)                 | NOLONGNAME                | NOOFFSET                 | NOSHOWINC               |
| NODECK                  | NOGONUMBER              | NOLSEARCH                 | NOOPTIMIZE               | SOURCE                  |
| NODLL <sup>3</sup>      | HALT(16) <sup>1</sup>   | MAXMEM(2000) <sup>5</sup> | PLIST(HOST) <sup>1</sup> | SPILL(128) <sup>5</sup> |
| NOSSCOMM                | START                   | TARGET(LE)                | TERMINAL                 | NOTEST <sup>6</sup>     |
| NOUPCONV                | XREF                    |                           |                          |                         |

Table 60. C language options (continued)

1. This option is used by IBM C/C++ for MVS/ESA V3R2 and subsequent releases
2. NOPTRACE, PPCHECK, GOTO, ACCURACY, PARM, NOENUM, NOEXTERN, TRUNC, INIT, NOPORT, GENERAL
3. This option is used by IBM C/C++ for MVS/ESA V3R1 and subsequent releases
4. AUTO, NOREPORT, 100, 1000
5. This option is used only by IBM AD/Cycle C/370 V1R2
6. SYM, BLOCK, LINE, NOPATH

Table 61. C++ language options

|             |                       |                           |                         |                     |
|-------------|-----------------------|---------------------------|-------------------------|---------------------|
| ARGPARSE    | NOIDL <sup>1</sup>    | NESTINC(255) <sup>2</sup> | NOSHOWINC               | TERMINAL            |
| NOATTRIBUTE | NOINFO                | OBJECT                    | NOSOM                   | NOTEST              |
| NOCSECT     | NOINLRPT              | NOOE <sup>2</sup>         | SOMEINIT                | XREF                |
| EXECOPS     | LANGLVL(EXTENDED)     | NOOFFSET                  | NOSOMGS                 | HALT(16)            |
| NOEXPMAC    | NOLIST                | OPTIMIZE(0)               | SOURCE                  | MEMORY <sup>2</sup> |
| NOEXPORTALL | NOLOCALE <sup>2</sup> | PLIST(HOST) <sup>2</sup>  | NOSRCMSG                | NOSEQUENCE          |
| FLAG(I)     | LONGNAME              | NOPONLY                   | START <sup>2</sup>      | TEMPINC             |
| NOGONUMBER  | MARGINS               | REDIR                     | TARGET(LE) <sup>2</sup> |                     |

1. This option is used by IBM C/C++ for MVS/ESA V3R1 and subsequent releases.
2. This option is used by IBM C/C++ for MVS/ESA V3R2 and subsequent releases.

The installation CLIST customizes C++ compiler parameters in sample job DSNTEJ2E if you have specified C++ for MVS/ESA V3R2 or a subsequent release on panel DSNTIPU.

If you wish to run your samples applications with C++ for MVS/ESA V3R2 or a subsequent release, 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 62. If you have a problem executing the DB2 PL/I samples, ensure that your compiler options are consistent with these options.

Table 62. PL/I options

|                    |               |          |                 |             |
|--------------------|---------------|----------|-----------------|-------------|
| CHARSET(60,EBCDIC) | LINECOUNT(55) | LMESSAGE | MARGINS(2,72,0) | NOAGGREGATE |
| NOATTRIBUTES       | NOCOMPILE(S)  | NOCOUNT  | NOESD           | NOFLOW      |



Table 62. PL/I options (continued)

|              |          |            |             |                 |
|--------------|----------|------------|-------------|-----------------|
| 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 Enterprise PL/I for z/OS and OS/390 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 *DB2 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 63.

Table 63. DSNTEJ0 return codes

| Step    | PROCSTEP | Return code         |
|---------|----------|---------------------|
| PH00S01 |          | 0000, 0008, or 0012 |
| PH00S02 |          | 0000, 0004, or 0008 |
| PH00S03 | DSNUPROC | 0004, or 0008       |
| PH00S04 |          | 0000, 0004, or 0008 |
| PH00S05 |          | 0000, 0004, or 0008 |
| PH00S06 |          | 0000, 0004, or 0008 |
| PH00S07 |          | 0000, 0004, 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. *y* is either I or J.

The following access method services commands, which can be executed under TSO, delete the Version 7 sample data sets:

```
DELETE 'vcatalog.DSNDBD.DSN8D71A.*.I0001.*'  
DELETE 'vcatalog.DSNDBC.DSN8D61L.*.I0001.*'  
DELETE 'vcatalog.DSNDBD.DSN8D71P.*.y0001.*'  
DELETE 'vcatalog.DSNDBC.DSN8D71U.*.I0001.A001'  
DELETE 'vcatalog.DSNDB04.STAFF.I0001.A001'  
DELETE 'vcatalog.DSNDB04.TESTSTUF.I0001.A001'  
  
DELETE 'vcatalog.DSNDBC.DSN8D61P.*.J0001.A001'
```

## Phase 1: Creating and loading sample tables

This phase consists of three jobs: DSNTSJ1, DSNTSJ1L, and DSNTSJ1P. DSNTSJ1 invokes program DSNTIAD, which creates objects during the verification procedure. Run DSNTIEJ1 before running any other sample jobs.

DSNTSJ1L and DSNTSJ1P prepare and invoke program DSNTSJ2, which lists the contents of the sample tables. The difference between the jobs is that DSNTSJ1P requires the PL/I compiler and allows you to customize DSNTSJ2.

### Job DSNTSJ1

Job DSNTSJ1 consists of the following steps:

Table 64. Steps in job DSNTSJ1

| Step | Function                                                                                                                                                                                                                                                                                                                                                                                        |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1-4  | Creates all objects (storage group, databases, table spaces, tables, indexes, and views) used by the samples.                                                                                                                                                                                                                                                                                   |
| 5    | Drops synonyms                                                                                                                                                                                                                                                                                                                                                                                  |
| 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 DSNHASM is not commented out.                                                                                                 |

Table 64. Steps in job DSNTEJ1 (continued)

| Step  | Function                                                                                                      |
|-------|---------------------------------------------------------------------------------------------------------------|
| 11    | Creates the sample utility list.                                                                              |
| 12    | Loads the programming-related tables using the LOAD utility.                                                  |
| 13    | Loads the sample tables using the LOAD utility.                                                               |
| 14    | Checks data for referential integrity.                                                                        |
| 15    | Establishes a quiesce point using both log and image copies.                                                  |
| 16    | Makes an image copy of all the sample tables using the copy facility.                                         |
| 17    | Establishes another quiesce point using only image copies.                                                    |
| 18    | Reorganizes a table space and compiles statistics on all table spaces using the REORG and RUNSTATS utilities. |
| 19    | Performs a REORG TABLESPACE with SHRLEVEL CHANGE                                                              |
| 20    | Loads the sample tables using the LOAD utility.                                                               |
| 21    | Set CURRENT RULES register and add a check constraint using ALTER TABLE.                                      |
| 22    | Checks data for referential integrity.                                                                        |
| 23    | Checks data for check integrity.                                                                              |
| 24-27 | Performs the operations in steps 15-18 except for the REORG on partition 3 of the Employee table space.       |
| 28    | Unloads data from a partitioned table.                                                                        |

If DSNTEJ1 runs successfully, it produces the return codes shown in Table 65.

Table 65. DSNTEJ1 return codes

| Step    | PROCSTEP          | Return Code          |
|---------|-------------------|----------------------|
| PH01S01 |                   | 0000                 |
| PH01S02 |                   | 0000                 |
| PH01S03 |                   | 0000                 |
| PH01S04 |                   | 0000                 |
| PH01S05 |                   | 0000                 |
| PH01S06 |                   | 0000                 |
| PH01S07 | ASM<br>LKED       | 0000<br>0000         |
| PH01S08 | ASM<br>LKED       | 0000<br>0000         |
| PH01S09 | ASM<br>LKED       | 0000<br>0000         |
| PH01S10 | PC<br>ASM<br>LKED | 0004<br>0000<br>0000 |
| PH01S11 | IEBGENER          | 0000                 |
| PH01S12 | DSNUPROC          | 0000                 |
| PH01S13 | DSNUPROC          | 0004                 |
| PH01S14 | DSNUPROC          | 0000                 |
| PH01S15 | DSNUPROC          | 0000                 |

Table 65. *DSNTEJ1* return codes (continued)

| Step    | PROCSTEP | Return Code  |
|---------|----------|--------------|
| PH01S16 | DSNUPROC | 0000         |
| PH01S17 | DSNUPROC | 0000         |
| PH01S18 | DSNUPROC | 0000 or 0004 |
| PH01S19 | DSNUPROC | 0000         |
| PH01S20 | DSNUPROC | 0004         |
| PH01S21 |          | 0004         |
| PH01S22 | DSNUPROC | 0004         |
| PH01S23 |          | 0000         |
| PH01S24 | DSNUPROC | 0000         |
| PH01S25 | DSNUPROC | 0000         |
| PH01S26 | DSNUPROC | 0000         |
| PH01S27 | DSNUPROC | 0000         |
| PH01S28 | 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 DSNTEJ1L

DSNTEJ1L link-edits the DSNTEP2 object deck (DSNTEP2L) to create an executable load module DSNTEP2. DSNTEP2 is described in Appendix D of *DB2 Utility Guide and Reference*. DSNTEJ1L also binds and runs 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. Job DSNTEJ1L requires the LE link-edit and runtime libraries. DSNTEJ1L does not require the PL/I compiler.

If DSNTEJ1L runs successfully, it produces the return codes shown in Table 66.

Table 66. *DSNTEJ1L* return codes

| Step     | PROCSTEP | Return code  |
|----------|----------|--------------|
| PH01PS01 |          | 0000         |
| PH01PS02 |          | 0000 or 0004 |

You can compare the output from this job with the sample output for DSNTEJ1L found in member DSN8TJ1L in your *prefix*.SDSNIVPD data set.

If you run DSNTEJ1P before DSNTEJ1L, you can expect PH01PS02 of DSNTEJ1L to produce a return code of 0004 and the following message:

```
SQLWARNING ON GRANT  COMMAND, EXECUTE FUNCTION
RESULT OF SQL STATEMENT:
DSNT404I SQLCODE = 562, WARNING:  A GRANT OF A PRIVILEGE WAS IGNORED
      BECAUSE THE GRANTEE ALREADY HAS THE PRIVILEGE FROM THE GRANTOR
```

If either DSNTSJ1 or DSNTSJ1L 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 DSNTSJ1, run DSNTSJ0 to drop the sample data. If you rerun DSNTSJ1L, rerun it from the last successful step.

## Job DSNTSJ1P

If you have run DSNTSJ1L, you do not need to run DSNTSJ1P because they produce the same results. The major difference is that DSNTSJ1P uses the PL/I compiler and allows you to customize DSNTSJ2. DSNTSJ2 is described in Appendix D of *DB2 Utility Guide and Reference*. DSNTSJ1P precompiles, compiles, and link-edits PL/I program DSNTSJ2. 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 DSNTSJ1P runs successfully, it produces the return codes shown in Table 67.

Table 67. DSNTSJ1P return codes

| Step     | PROCSTEP | Return code  |
|----------|----------|--------------|
| PH01PS01 | PPLI     | 0000         |
|          | PC       | 0000         |
|          | PLI      | 0004         |
|          | LKED     | 0000         |
| PH01PS02 |          | 0000 or 0004 |

You can compare the output from this job with the sample output for DSNTSJ1P found in member DSN8TJ1P in your *prefix*.SDSNIVPD data set.

If you run DSNTSJ1L before DSNTSJ1P, you can expect PH01PS02 of DSNTSJ1P to produce a return code of 0004 and the following message:

```
SQLWARNING ON GRANT  COMMAND, EXECUTE FUNCTION
  RESULT OF SQL STATEMENT:
    DSNT404I  SQLCODE = 562, WARNING:  A GRANT OF A PRIVILEGE WAS IGNORED
      BECAUSE THE GRANTEE ALREADY HAS THE PRIVILEGE FROM THE GRANTOR
```

If either DSNTSJ1 or DSNTSJ1P 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 DSNTSJ1, run DSNTSJ0 to drop the sample data. If you rerun DSNTSJ1P, rerun it from the last successful step.

## Job DSNTEJ1U

DSNTEJ1U creates a database, table space, and table with CCSID Unicode. DSNTEJ1U loads data into the table from a data set containing a full range of characters in an EBCDIC Latin-1 code page, which results in a mix of single and double-byte characters in the Unicode table. It then runs DSN1PRNT on the table to dump the hex image, revealing which characters are stored in single bytes and which in double bytes.

Do not run this job unless your system has the prerequisite operating system level for Version 7 Unicode support. For more information, see "Function-Dependent Program Requirements" in the Version 7 DB2 Program Directory.

If DSNTEJ1U runs successfully, it produces the return codes shown in Table 68.

Table 68. DSNTEJ1U return codes

| Step     | PROCSTEP | Return code |
|----------|----------|-------------|
| PH01US01 |          | 0000        |
| PH01US02 |          | 0000        |
| PH01US03 | DSNUPROC | 0000        |
| PH01US04 |          | 0000        |

## 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 DSNTEJ2A

DSNTEJ2A 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. DSNTEJ2A then uses the LOAD utility to put data into copies of the unloaded tables. DSNTIAUL is described in Appendix D of *DB2 Utility Guide and Reference*.

Table 69. DSNTEJ2A 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 for DSNTEJ2A found in member DSN8TJ2A in your *prefix*.SDSNIVPD data set.

## 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 405.

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 355. 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 70.

Table 70. DSNTEJ2C return codes

| Step     | PROCSTEP           | Return code  |
|----------|--------------------|--------------|
| PH02CS01 |                    | 0000         |
| PH02CS02 | PC                 | 0004         |
|          | COB/COB2/IBMCOB    | 0000 or 0004 |
|          | PLKED <sup>1</sup> | 0004         |
|          | LKED               | 0000 or 0004 |
| PH02CS03 | PC                 | 0000         |
|          | COB/COB2/IBMCOB    | 0000 or 0004 |
|          | PLKED <sup>1</sup> | 0004         |
|          | LKED               | 0000         |
| PH02CS04 |                    | 0000         |

<sup>1</sup>Only for IBM COBOL

You can compare the output from this job with the sample output for DSNTEJ2C found in member DSN8TJ2C in your *prefix*.SDSNIVPD data set.

## 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 365.

If DSNTEJ2D runs successfully, it produces the return codes shown in Table 71.

Table 71. DSNTE2D return codes

| Step     | PROCSTEP | Return code  |
|----------|----------|--------------|
| PH02DS01 | PC       | 0004         |
|          | C        | 0000         |
|          | PLKED    | 0000 or 0004 |
|          | LKED     | 0004         |

Table 71. DSNTJ2D return codes (continued)

| 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 for DSNTJ2D found in member DSN8TJ2D in your *prefix*.SDSNIVPD data set.

## Job DSNTJ2E

Job DSNTJ2E 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 DSNTJ2C” on page 365.

If you do not have C++ installed, skip steps PH02US08 and PH02US09. Also remove all statements that refer to DAYNAME and MONTHNAME from part DSNTJ2E in the *prefix*.SDSNSAMP library.

If DSNTJ2E runs successfully, it produces the return codes shown in Table 72.

Table 72. DSNTJ2E 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 for DSNTJ2E found in member DSN8TJ2E in your *prefix*.SDSNIVPD data set.

## Job DSNTJ2F

Job DSNTJ2F tests the Fortran program preparation procedures. The FORTRAN job runs only the phone application. See the phone application description under DSNTJ2C on page 365.

If DSNTJ2F runs successfully, it produces the return codes shown in Table 73 on page 367.



Table 73. DSNTJ2F 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 for DSNTJ2F found in member DSN8TJ2F in your *prefix*.SDSNIVPD data set.

## Job DSNTJ2P

Job DSNTJ2P tests the PL/I program preparation procedures. The PL/I job runs the phone application.

If DSNTJ2P runs successfully, it produces the return codes shown in Table 74.

Table 74. DSNTJ2P 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 for DSNTJ2P found in member DSN8TJ2P in your *prefix*.SDSNIVPD data set.

## Job DSNTJ2U

DSNTJ2U prepares and tests several sample user-defined functions, as well as a driver program to exercise them.

In order for the Install CLIST to generate Job DSNTJ2U, you must complete the following steps during installation:

- Specify that the host has access to C/C++ for OS/390 Version 1 Release 3 or a subsequent release on Install Panel DSNTIPU
- Specify the name of the default WLM environment on Install panel DSNTIPX

The sample user-defined functions are:

| Function      | Description                                                                                                       |
|---------------|-------------------------------------------------------------------------------------------------------------------|
| <b>ALTDAT</b> | Returns the current date in a user-specified format or converts a user-specified date from one format to another. |

|                       |                                                                                                                                                                               |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ALTIME</b>         | Returns the current time in a user-specified format or converts a user-specified time from one format to another.                                                             |
| <b>CURRENCY</b>       | Formats a floating point number as a currency value.                                                                                                                          |
| <b>DAYNAME</b>        | Returns the day of the week for a user-specified date in ISO format.                                                                                                          |
| <b>MONTHNAME</b>      | Returns the month for a user-specified date in ISO format.                                                                                                                    |
| <b>TABLE_LOCATION</b> | Returns the location name of a table, view, or undefined object found after resolving aliases for a user-specified object. See <i>DB2 SQL Reference</i> for more information. |
| <b>TABLE_NAME</b>     | Returns the name of a table, view, or undefined object found after resolving aliases for a user-specified object. See <i>DB2 SQL Reference</i> for more information.          |
| <b>TABLE_SCHEMA</b>   | Returns the schema name of a table, view, or undefined object found after resolving aliases for a user-specified object. See <i>DB2 SQL Reference</i> for more information.   |
| <b>WEATHER</b>        | Returns sample weather data obtained from a TSO data set by way of demonstrating the usefulness of a user-defined function table function.                                    |

For more information about user-defined functions, see Part 2 of *DB2 Application Programming and SQL Guide*.

Job DSNTEJ2U consists of the following steps:

Table 75. Steps in job DSNTEJ2U

| Step | Function                                                                                                                                                                                                                 |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | Drops all specific sample user-defined functions.                                                                                                                                                                        |
| 2    | Creates and registers all sample user-defined functions. Grants EXECUTE authority to PUBLIC for the sample user-defined functions.                                                                                       |
| 3-10 | Prepares the eight external programs used by specific user-defined functions.                                                                                                                                            |
| 11   | Binds the package for the TABLE_NAME, TABLE_SCHEMA, and TABLE_LOCATION functions. These are the only sample functions that issue SQL statements. This step also grants EXECUTE authority on these two samples to PUBLIC. |
| 12   | Invokes DSNTEP2 to exercise the sample user-defined functions.                                                                                                                                                           |
| 13   | Prepares DSN8DUWF, the external module for the sample user defined table function, WEATHER.                                                                                                                              |
| 14   | Prepares DSN8DUWC, a sample client function for statically invoking the WEATHER user-defined table function.                                                                                                             |
| 15   | Binds the package and plan for DSN8DUWC and grants the necessary authorities.                                                                                                                                            |
| 16   | Invokes DSN8DUWC.                                                                                                                                                                                                        |

If DSNTEJ2U runs successfully, it produces the return codes shown in Table 76 on page 369.

Table 76. DSNTSJ2U return codes

| Step                | PROCSTEP | Return code  |
|---------------------|----------|--------------|
| PH02US01            |          | 0000         |
| PH02US02            |          | 0000 or 0004 |
| PH02US03 - PH02US07 | PC       | 0004         |
|                     | C        | 0000         |
|                     | PLKED    | 0004         |
|                     | LKED     | 0000         |
| PH02US08 - PH02US09 | PC       | 0004         |
|                     | CP       | 0000         |
|                     | PLKED    | 0004         |
|                     | LKED     | 0000         |
| PH02US10            | PC       | 0000         |
|                     | C        | 0000         |
|                     | PLKED    | 0004         |
|                     | LKED     | 0000         |
| PH02US11            |          | 0000         |
| PH02US12            |          | 0004         |
| PH02US13            | PC       | 0004         |
|                     | C        | 0000         |
|                     | PLKED    | 0004         |
|                     | LKED     | 0000         |
| PH02US14            | PC       | 0000         |
|                     | C        | 0000         |
|                     | PLKED    | 0004         |
|                     | LKED     | 0000         |
| PH02US15            |          | 0000 or 0004 |
| PH02US16            |          | 0000         |

You can compare the output from this job with the sample output for DSNTSJ2U found in member DSN8TJ2U in your *prefix*.SDSNIVPD data set.

## 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.

DSNTESC, the sample PLAN\_TABLE, creates a Version 7 PLAN\_TABLE with an index optimized for access path hints. It also demonstrates how to migrate your PLAN\_TABLEs from earlier releases of DB2. If you are migrating from Version 6, DSNTESC migrates your Version 6 sample PLAN\_TABLE to Version 7. Migration of the sample PLAN\_TABLE from a prior release, should not be performed more than once. DSNTESC also includes SQL to create a sample DSN\_STATEMENT\_TABLE and a sample DSN\_FUNCTION\_TABLE.

SPUFI (SQL Processor Using File Input) is a facility of DB2I. "Testing SPUFI" on page 370 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. On the DB2I defaults panel, change the DB2 name to the DB2 subsystem name you entered on panel DSNTIPM during installation. Then select DB2I on the ISPF Primary Option Menu.
4. Select SPUFI on the DB2I menu.
5. Enter the library name '*prefix*.NEW.SDSNSAMP(DSNTESA)' 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 DSNTESA 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 short-lived table space and table (as discussed in "Dynamic SQL statements (DSNTESA, DSNTESQ)" on page 421)
- Once with member DSNTESC, which creates a table space and creates a PLAN\_TABLE, DSN\_FUNCTION\_TABLE, and DSN\_STATEMENT\_TABLE in that table space (as discussed in Part 5 (Volume 2) of *DB2 Administration Guide*). If migrating from DB2 Version 6, the installation CLIST tailors DSNTESC to migrate your Version 6 PLAN\_TABLE to Version 7 using a schema name of DSN8510. If your Version 6 PLAN\_TABLE has a different schema name, edit DSNTESC to use your schema name.
- Once with member DSNTESQ, 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 must drop either a Version 6 or Version 7 PLAN\_TABLE, remove the appropriate comments from the job to issue the DROP statements.

Also, make sure that the user ID 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 database 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
*.DSNESPCS.DSNESM68

```

The package list for the DSNESPRR plan is as follows:

```

location_name.PRIVATRR.DSNESM68
*.DSNESPRR.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 DSNTJ1 prepares the assembler interface, and job DSNTJ3C prepares the COBOL interface. The connection manager program and the phone application each exist in COBOL and PL/I. Job DSNTJ3C prepares the COBOL version; job DSNTJ3P 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 DSNTJ3C 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 355. Remember that you must use the same type of COBOL to prepare DSNTJ2C, DSNTJ3C, DSNTJ4C, and DSNTJ5C.

## Jobs DSNTJ3C and DSNTJ3P:

To prepare for the distributed sample application, DSNTJ3C 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 7.

Because DSNTSJ3C does a remote bind, you must set up your local and remote systems for remote communication before running this job. The sample jobs DSNTSJ1 and DSNTSJ6 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 13. Connecting distributed database systems” on page 465.

If DSNTSJ3C runs successfully, it produces the return codes shown in Table 77.

*Table 77. DSNTSJ3C return codes*

| Step     | PROCSTEP           | Return code  |
|----------|--------------------|--------------|
| PH03CS01 | PC                 | 0004         |
|          | COB/COB2/IBMCOB    | 0000 or 0004 |
|          | PLKED <sup>1</sup> | 0004         |
|          | LKED               | 0000         |
| PH03CS02 | PC                 | 0004         |
|          | COB/COB2/IBMCOB    | 0000 or 0004 |
|          | PLKED <sup>1</sup> | 0004         |
|          | LKED               | 0000         |
| PH03CS03 | PC                 | 0000         |
|          | COB/COB2/IBMCOB    | 0004         |
|          | PLKED <sup>1</sup> | 0004         |
|          | LKED               | 0000         |
| PH03CS04 | PC                 | 0000 or 0004 |
|          | COB/COB2/IBMCOB    | 0000 or 0004 |
|          | PLKED <sup>1</sup> | 0004         |
|          | LKED               | 0000         |
| PH03CS05 |                    | 0000         |
| PH03CS06 |                    | 0000 or 0004 |

<sup>1</sup>Only for IBM COBOL

Step PH03CS06 can give a return code of 0004 if sample job DSNTSJ1 was not run on the remote system. For testing, you should run job DSNTSJ1 on the remote system. If DSNTSJ3P runs successfully, it produces the return codes shown in Table 78.

*Table 78. DSNTSJ3P 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        |

The three steps in job DSNTEJ3P, steps PH03PS01, PH03PS02, and PH03PS03, prepare the ISPF/CAF sample application.

## 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 247 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 sample applications panel, shown in Figure 44.

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)    |
| 4 | C EMPLOYEE RESUME  | (DB2 ISPF C Application)        |
| 5 | C EMPLOYEE PHOTO   | (DB2 ISPF & GDMM C Application) |

SPECIFY DB2 SUBSYSTEM NAME ====> DSN

PRESS: END TO EXIT

Figure 44. Initial panel of the ISPF/CAF application

Choosing option 1 or 2 on the sample applications panel during Phase 3 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 405. Choosing option 3 on the sample applications panel during Phase 6 invokes the COBOL organization application, which uses DRDA access to distributed data. For more information, see “The distributed organization application scenario” on page 409. Choosing options 4 and 5 on the sample applications panel during Phase 7 invokes the “Employee Resume” and “Employee Photo” applications, which processes LOB data. You must run the phase 7 sample jobs before you can access options 4 and 5 on the sample applications panel. For more information, see “Employee resume and photo scenarios” on page 417.

---

## 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 405.

## Jobs DSNTSJ4C and DSNTSJ4P

Job DSNTSJ4C is for COBOL; DSNTSJ4P 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. **Recommendation:** Use 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 355. Remember, you must use the same type of COBOL to prepare DSNTSJ2C, DSNTSJ3C, DSNTSJ4C, and DSNTSJ5C.

If DSNTSJ4C runs successfully, it produces the return codes shown in Table 79 .

Table 79. DSNTSJ4C return codes

| Step     | PROCSTEP           | Return code |
|----------|--------------------|-------------|
| PH04CS01 | PC                 | 0004        |
|          | COB/COB2/IBMCOB    | 0000        |
|          | PLKED <sup>1</sup> | 0004        |
|          | LKED               | 0004        |
| PH04CS02 | PC                 | 0000        |
|          | COB/COB2/IBMCOB    | 0000        |
|          | PLKED <sup>1</sup> | 0004        |
|          | LKED               | 0004        |
| PH04CS03 | PC                 | 0000        |
|          | COB/COB2/IBMCOB    | 0000        |
|          | PLKED <sup>1</sup> | 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        |

<sup>1</sup>Only for IBM COBOL



For DSNTEJ4C, the warning code expected from the precompiler step PH04CS01 is:

```
DB2 SQL PRECOMPILER      MESSAGES
DSNH0531 W   NO SQL STATEMENTS WERE FOUND
```

If DSNTEJ4P runs successfully, it produces the return codes shown in Table 80.

*Table 80. 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         |

Table 80. DSNTEJ4P return codes (continued)

| Step     | PROCSTEP | Return Code |
|----------|----------|-------------|
| 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 45 is displayed.

```
MAJOR SYSTEM ...: 0      ORGANIZATION
ACTION .....:
OBJECT .....:
SEARCH CRITERIA.:
DATA .....:
```

Figure 45. Organization version of FORMAT command display

When the following command is entered, the panel shown in Figure 46 is displayed.

- /FORMAT DSN8IPFO, which starts the PL/I projects version.

```
MAJOR SYSTEM ...: P      PROJECTS
ACTION .....:
OBJECT .....:
SEARCH CRITERIA.:
DATA .....:
```

Figure 46. 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 47 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 47. 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 DSNTSJ5A

DSNTSJ5A 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. If you are using CICS Version 4 or CICS Transaction Server 1.1, you need to modify job DSNTSJ5A to use steps DSN8FRCT and DSN8FRDO. You may need to tailor step DSN8FRDO for your system.

Table 81. DSNTSJ5A return codes

| Step     | PROCSTEP | Return code |
|----------|----------|-------------|
| PH05AS01 |          | 0000        |
| PH05AS02 |          | 0000        |
| PH05AS03 |          | 0000        |

### Jobs DSNTSJ5C and DSNTSJ5P

Job DSNTSJ5C installs the sample application transactions in COBOL and prepares the organization application. Job DSNTSJ5P 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 355. Remember, you must use the same type of COBOL to prepare DSNTEJ2C, DSNTEJ3C, DSNTEJ4C, and DSNTEJ5C.

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 DSNTEJ5C runs successfully, it produces the return codes shown in Table 82.

*Table 82. DSNTEJ5C return codes*

| Step  | PROCSTEP | Return code  |
|-------|----------|--------------|
| MAPG  | ASSEM    | 0000         |
| MAPD  | ASSEM    | 0000         |
| DSNH  |          | 0000 or 0004 |
| BIND  |          | 0000         |
| MAPGP | ASSEM    | 0000         |
| MAPGL |          | 0000         |
| MAPDP | ASSEM    | 0000         |
| MAPDL |          | 0000         |

If DSNTEJ5C fails or abends, rerun the job from the last successful step. To receive more prepare-time detail from DSNTEJ5C, 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 *DB2 Command Reference* for more information.

If DSNTEJ5P runs successfully, it produces the return codes shown in Table 83.

*Table 83. DSNTEJ5P 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 83. DSNTJEJ5P return codes (continued)

| 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 or 0004 |
| 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         |

Table 83. DSNTJ5P return codes (continued)

| Step     | PROCSTEP | Return code |
|----------|----------|-------------|
| PH05PS36 | ASSEM    | 0000        |
| PH05PH37 |          | 0000        |

If DSNTJ5P fails or abends, rerun the job from the last successful step. You might find it convenient to break up DSNTJ5P 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 48 and Figure 49 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 48. 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 49. Initial panel for the project application in CICS

Refer to “Specifying values in the sample application panels” on page 396 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 DSNTJ5C or job DSNTJ5P. 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 , DSNTEJ6V, and DSNTEJ6Z)
- The stored procedure for invoking WLM\_REFRESH (DSNTEJ6W)
- The stored procedure for IMS Open Database Access (DSNTEJ61 and DSNTEJ62)
- The SQL procedure batch sample (DSNTEJ63 and DSNTEJ64)
- The SQL procedures processor invocation sample (DSNTEJ65)

The installation CLIST prepares samples DSNTEJ6, DSNTEJ6S, DSNTEJ6P, DSNTEJ6T, DSNTEJ6D, DSNTEJ63, DSNTEJ64, and DSNTEJ65 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, DSNTEJ6W, DSNTEJ6Z, DSNTEJ6D, DSNTEJ63, DSNTEJ64, and DSNTEJ65 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 DSNTEJ6, DSNTEJ62, DSNTEJ6V, and DSNTEJ65 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 only customizes phase 6 sample jobs DSNTSJ63, DSNTSJ64, DSNTSJ65, and DSNTSJ6U, and DSNTSJ6V. Jobs DSNTSJ63, DSNTSJ64, and DSNTSJ65 use the local location name when the REMOTE LOCATION field is blank. Jobs DSNTSJ6U and DSNTSJ6V do not use a remote location name.
- 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), the stored procedures samples are prepared and customized for local use. However, the DRDA access sample is not prepared. This includes DSNTSJ6 and the DRDA access component of DSNTSJ3C.
- 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 that 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 DSNTSJ3C. Before this application can be run correctly as a DRDA access sample, you must run job DSNTSJ6 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 DSNTSJ6 at the client site only; do **not** run the client version of DSNTSJ6 at the server.
- Edit the server version of DSNTSJ6 at the remote server site; do **not** run the server version of DSNTSJ6 at the client.
- Locate the following text in the server version of DSNTSJ6 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 DSNTSJ6

Job DSNTSJ6 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 DSNTSJ6 runs successfully, it produces the return code shown in Table 84 on page 383.



Table 84. DSNTSJ6 return codes

| Step    | PROCSTEP | Return code |
|---------|----------|-------------|
| PH06S01 |          | 0000        |

After job DSNTSJ6 has completed successfully, start the distributed application scenario by following the instructions in “Starting an application in an ISPF/TSO environment in phase 6” on page 392.

## 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 DSN8710.VPHONE and DSN8710.VEMPLP at a remote location |

### Notes:

1. It is assumed that the views DSN8710.VPHONE and DSN8710.VEMPLP and their underlying tables exist at the remote location. If they do not exist, run job DSNTSJ1 to create them.
2. Step 1 always has one set of SQL statements that fail; it either drops the views or the aliases, but not both.
3. 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.

If you specified DRDA as the default database protocol on panel DSNTIP5, you must specify DBPROTOCOL(PRIVATE) to test DB2 private protocol access. After running this job, re-run Phase 2 jobs DSNTSJ2C through DSNTSJ2P. 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(DSNTIA71) PARM('RC0') -
      LIB('prefix.RUNLIB.LOAD')
END
//SYSIN DD *
DROP VIEW DSN8710.VPHONE;
DROP VIEW DSN8710.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(DSNTIA71) -
        LIB('prefix.RUNLIB.LOAD')
  END
//SYSIN DD *
  CREATE ALIAS VPHONE FOR SAMPLOC.DSN8710.VPHONE;
  CREATE ALIAS VEMPLP FOR SAMPLOC.DSN8710.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
- One sample using the utilities stored procedure DSNUTILS
- One sample for IMS Open Database Access (ODBA) support
- Two samples for SQL procedures
- One sample to list the settings of subsystem parameters
- One sample to invoke a stored procedure that refreshes the WLM environment

Most of the applications prepare and run two programs; one prepares a stored procedure, and one executes a client program that calls the stored procedure and returns some response.

The stored procedure sample jobs are edited only when you specify selected fields on the installation panels. For details on which fields are required for the sample jobs, see the notes in Table 56 on page 350. To run these applications, you must first start the DB2-established or WLM-established stored procedures address space (SPAS). See “Routine parameters panel: DSNTIPX” on page 214 for information on generating the JCL to start the SPAS. Before starting the SPAS, you must have Language Environment and a Language Environment-compatible version of PL/I, C, or COBOL installed (depending on the job) to run the stored procedure sample jobs.

## Stored procedure sample without result set

This application consists of two jobs: DSNTSJ6S and DSNTSJ6P. Job DSNTSJ6S must be run before job DSNTSJ6P. To run these jobs, you must have PL/I for MVS installed on your client and server systems in addition to Language Environment.

This application prepares and runs:

- 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 in a parameter that is passed back, and prints the results.

For concurrent installations at two DB2 locations:

- Run the server version of DSNTSJ6S on the server system only; do **not** run the client version of DSNTSJ6S on the server
- Run the client version of DSNTSJ6P on the client system only; do **not** run the server version of DSNTSJ6P on the client

## Job DSNTSJ6S

Job DSNTSJ6S compiles and link-edits the sample stored procedure DSN8EP2. It also updates the SYSIBM.SYSROUTINES 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 DSNTSJ6S at the DB2 server location.

If DSNTSJ6S runs successfully, it produces the return codes shown in Table 85.

Table 85. DSNTSJ6S return codes

| Step     | PROCSTEP | Return code |
|----------|----------|-------------|
| PH06SS01 |          | 0000        |
| PH06SS02 |          | 0000        |
| PH06SS03 | PPLI     | 0000        |
|          | PC       | 0004        |
|          | PLI      | 0004        |
|          | LKED     | 0000        |

If SQLCODE -592 is received during the execution of DSNTSJ6T, ensure that SPA is active.

## Job DSNTSJ6P

Job DSNTSJ6P compiles, link-edits, binds, and runs a sample program, DSN8EP1, that invokes the sample stored procedure.

Before you run DSNTSJ6P, run DSNTSJ6S to create the sample stored procedure.

If DSNTSJ6P runs successfully, it produces the return codes shown in Table 86.

Table 86. DSNTSJ6P 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 DSNTSJ6P lists each DB2 command executed, followed by the messages generated by the DB2 command processor.

You can compare the output from this job with the sample output for DSNTSJ6P found in member DSN8TJ6P in your *prefix.SDSNIVPD* data set.

## Stored procedure sample with result set

This application consists of two jobs; DSNTSJ6T and DSNTSJ6D. Job DSNTSJ6T must be run before job DSNTSJ6D. You must have C for MVS installed, in addition to Language Environment to run DSNTSJ6D and DSNTSJ6T. This application prepares and runs:

- 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 in a result set, and prints the results.

For concurrent installations at two DB2 locations:

- Run the server version of job DSNTSJ6T on the server side only; do **not** run it on the client side
- Run the client version of job DSNTSJ6D on the client side only; do **not** run it on the server side

## Job DSNTSJ6T

Job DSNTSJ6T registers, prepares, and binds the sample stored procedure, DSN8ED2, on the server. It also defines a created temporary table to receive the IFI output that is returned as a result set.

If DSNTSJ6T runs successfully, it produces the return codes shown in Table 87.

Table 87. DSNTSJ6T return codes

| Step     | PROCSTEP | Return code  |
|----------|----------|--------------|
| PH06TS01 |          | 0000         |
| PH06TS02 |          | 0000         |
| PH06TS03 | PC       | 0004         |
|          | C        | 0000         |
|          | PLKED    | 0004         |
|          | LKED     | 0000         |
| PH06TS04 |          | 0000 or 0004 |

If SQLCODE -592 is received during the execution of DSNTSJ6T, ensure that SPA is active.

## Job DSNTSJ6D

Job DSNTSJ6D compiles, link-edits, binds, and runs sample program DSN8ED1, that invokes the sample for using stored procedure result sets.

Before you run DSNTSJ6D, run DSNTSJ6T to create the sample stored procedure.

If DSNTSJ6D runs successfully, it produces the return codes shown in Table 88 on page 387.

Table 88. DSNTEJ6D return codes

| Step     | PROCSTEP | Return code  |
|----------|----------|--------------|
| PH06DS01 | PC       | 0000         |
|          | C        | 0000 or 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.

You can compare the output from this job with the sample output for DSNTEJ6D found in member DSN8TJ6D in your *prefix*.SDSNIVPD data set.

## Sample utilities stored procedure

The DSNUTILS stored procedure enables execution of DB2 utilities from a DB2 application program using the SQL CALL statement. When called, DSNUTILS dynamically allocates the specified data sets, creates the utility input stream (SYSIN), invokes DB2 utilities (DSNUTILB), deletes all rows currently in the created temporary table (SYSIBM.SYSPRINT), captures the utility output stream (SYSPRINT), and puts this output into the created temporary table (SYSIBM.SYSPRINT).

The DSNUTILS stored procedure must run as a WLM-managed stored procedure.

## Job DSNTEJ6U

Job DSNTEJ6U compiles, link-edits, binds, and runs sample PL/I program DSN8EPU, which invokes the DSNUTILS stored procedure to execute a utility.

Before you run DSNTEJ6U, verify that the DSNUTILS stored procedure was successfully created in job DSNTIJSG. DSNTEJ6U requires a WLM procedure. See Appendix B of *DB2 Utility Guide and Reference* for an example of how to customize a WLM proc for DSNUTILS.

Table 89. DSNTEJ6U return codes

| Step     | PROCSTEP | Return code |
|----------|----------|-------------|
| PH06US01 | PC       | 0000        |
|          | PLI      | 0004        |
|          | LKED     | 0000        |
| PH06US02 |          | 0004        |
| PH06US03 |          | 0000        |

Output from a successful execution of DSNTEJ6U 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 *DB2 Utility Guide and Reference*.

You can compare the output from this job with the sample output for DSNTEJ6U found in member DSN8TJ6U in your *prefix*.SDSNIVPD data set.

## Job DSNTEJ6V

Job DSNTEJ6V compiles, link-edits, binds, and runs sample C++ program DSN8EE1, which invokes the DSNUTILS stored procedure to execute a utility. DSNTEJ6V requires a WLM procedure. See Appendix B of *DB2 Utility Guide and Reference* for an example of how to customize a WLM proc for DSNUTILS.

Table 90. DSNTEJ6V return codes

| Step     | PROCSTEP | Return code  |
|----------|----------|--------------|
| PH06VS01 | PC       | 0004         |
|          | CP       | 0000         |
|          | PLKED    | 0004         |
|          | LKED     | 0004         |
| PH06VS02 | PC1      | 0000         |
|          | CP1      | 0000         |
|          | PC2      | 0000         |
|          | CP2      | 0000         |
|          | PLKED    | 0004         |
|          | LKED     | 0000         |
| PH06VS03 |          | 0000 or 0004 |
| PH06VS04 |          | 0000         |

A successful execution of DSNTEJ6V unloads rows and columns from the PROJ sample table. For more details on utilities and invoking utilities as a stored procedure, see Appendix B of *DB2 Utility Guide and Reference*.

You can compare the output from this job with the sample output for DSNTEJ6V found in member DSN8TJ6V in your *prefix.SDSNIVPD* data set.

## Job DSNTEJ6W

Job DSNTEJ6W is a JCL job that creates and initializes a sample SAF resource profile, and prepares, binds, and executes DSN8ED6. DSN8ED6 is a C-language caller of WLM\_REFRESH that accepts the WLM environment name and, optionally, the subsystem ID and an SQLID to be in effect when WLM\_REFRESH is invoked.

The installation CLIST customizes DSNTEJ6W to run in and recycle the same WLM environment, which is the environment you specified in the WLM ENVIRONMENT field on installation panel DSNTIPX.

If you are not authorized to create special resource profiles, have your system security administrator perform the first step of this job. The SQLID used to run DSNTEJ6W needs READ access to this profile.

For more information on WLM, see *DB2 Application Programming and SQL Guide*.

Table 91. DSNTEJ6W return codes

| Step     | PROCSTEP | Return code |
|----------|----------|-------------|
| PH06WS01 |          | 0000        |
| PH06WS02 | PC       | 0000        |
|          | C        | 0000        |

Table 91. DSNTSJ6W return codes (continued)

| Step     | PROCSTEP | Return code  |
|----------|----------|--------------|
|          | PLKED    | 0004         |
|          | LKED     | 0000         |
| PH06WS03 |          | 0000 or 0004 |
| PH06WS04 |          | 0000         |

## Job DSNTSJ6Z

Job DSNTSJ6Z generates a report of current subsystem parameter settings. This report is generated by DSN8ED7, a C-language caller of stored procedure DSNWZP. DSNTSJ6Z requires a WLM procedure.

Table 92. DSNTSJ6Z return codes

| Step     | PROCSTEP | Return code  |
|----------|----------|--------------|
| PH06ZS01 | PC       | 0000         |
|          | C        | 0000         |
|          | PLKED    | 0004         |
|          | LKED     | 0000         |
| PH06ZS02 |          | 0000 or 0004 |
| PH06ZS03 |          | 0000         |

A successful execution of DSNTSJ6Z provides a report in the following format:

DSN8ED7: Sample DB2 Configuration Setting Report Generator

| Macro Name | Parameter Name | Current Setting | Description/<br>Install Field Name | Install<br>Panel ID | Fld<br>No. |
|------------|----------------|-----------------|------------------------------------|---------------------|------------|
| DSN6SYSP   | AUDITST        | 0000000000      | AUDIT TRACE                        | DSNTIPN             | 1          |
| DSN6SYSP   | CONDBAT        | 0000000064      | MAX REMOTE CONNECTED               | DSNTIPE             | 4          |
| DSN6SYSP   | CTHREAD        | 00030           | MAX USERS                          | DSNTIPE             | 2          |
| DSN6SYSP   | DLDREQ         | 00005           | LEVELID UPDATE FREQ                | DSNTIPL             | 14         |
| DSN6SYSP   | PCLOSEN        | 00005           | SWITCH CHKPTS                      | DSNTIPL             | 12         |

## Sample 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. For more information on IMS ODBA, see *DB2 Application Programming and SQL Guide*. ODBA requires IMS Version 6.

This application consists of two jobs: DSNTSJ61 and DSNTSJ62. Job DSNTSJ61 must be run before DSNTSJ62. You must have COBOL for MVS and VM and Language environment installed to run DSNTSJ61 and DSNTSJ62. You must start a WLM-established stored procedure address space to run DSNTSJ61 and DSNTSJ62. 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:

IMSVS.RESLIB

For more information on the setting up a WLM-established stored procedure address space see *DB2 Application Programming and SQL Guide*.

## Job DSNTEJ61

Job DSNTEJ61 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.

Before running DSNTEJ61, read the Dependencies section of the job prolog to verify that the server site is correctly configured. If DSNTEJ61 runs successfully, it produces the return codes shown in Table 93.

Table 93. DSNTEJ61 return codes

| Step     | PROCSTEP | Return code |
|----------|----------|-------------|
| PH061S01 |          | 0000        |
| PH061S02 |          | 0000        |
| PH061S03 | 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.

Before running DSNTEJ62, perform the manual editing described in the Dependencies section of the job prolog. If DSNTEJ62 runs successfully, it produces the return codes shown in Table 94.

Table 94. DSNTEJ62 return codes

| Step     | PROCSTEP | Return code  |
|----------|----------|--------------|
| PH062S01 | PC       | 0000         |
|          | COB      | 0000         |
|          | PLKED    | 0004         |
|          | LKED     | 0000         |
| PH062S02 |          | 0000 or 0004 |
| PH062S03 |          | 0000         |

## Sample SQL procedures

The two applications for SQL procedures are:

- DSNTEJ63 and DSNTEJ64

Job DSNTEJ63 must be run before DSNTEJ64. Job DSNTEJ63 prepares a sample SQL procedure. Job DSNTEJ64 prepares and executes a C program that calls the sample SQL procedure.

- DSNTEJ65

Job DSNTEJ65 has three parts:

- Prepares a C program that calls the DB2 SQL procedures processor (DSNTPSMP).



- Prepares a sample SQL procedure.
- Prepares and executes a C program that calls the SQL procedure.

Use DSNTEJ65 to verify that the DB2 SQL procedures processor, DSNTPSMP, is working correctly.

C and Language Environment are required for jobs DSNTEJ63, DSNTEJ64, and DSNTEJ65. For job DSNTEJ65, you must start a WLM-established stored procedures address space for DSNTPSMP. You must update the start-up procedure for the WLM-established stored procedure address space as listed in the prolog of sample start-up procedure DSN8WLMP. For more information on the setting up a WLM-established stored procedure address space, see Part 6 of *DB2 Application Programming and SQL Guide*.

## Job DSNTEJ63

Job DSNTEJ63 prepares the sample SQL procedure, DSN8ES1, which accepts a department number and returns salary and bonus data.

Before running DSNTEJ63, perform the manual editing described in the dependencies section of the job prolog. If DSNTEJ63 runs successfully, it produces the return codes shown in Table 95.

If you specified a remote location name on installation panel DSNTIPY, then DSNTEJ63 should be run on the remote server site.

Table 95. DSNTEJ63 return codes

| Step     | PROCSTEP | Return code  |
|----------|----------|--------------|
| PH063S01 |          | 0000         |
| PH063S02 | PC       | 0004         |
|          | PCC      | 0004         |
|          | C        | 0000         |
|          | PLKED    | 0004         |
|          | LKED     | 0000         |
| PH063S03 |          | 0000 or 0004 |

## Job DSNTEJ64

Job DSNTEJ64 prepares and executes DSN8ED3, a sample routine that calls the sample SQL procedure, DSN8ES1. You must run job DSNTEJ63 before running job DSNTEJ64.

Before running DSNTEJ64, perform the manual editing described in the dependencies section of the job prolog. If DSNTEJ64 runs successfully, it produces the return codes shown in Table 96.

Table 96. DSNTEJ64 return codes

| Step     | PROCSTEP | Return code  |
|----------|----------|--------------|
| PH064S01 | PC       | 0000         |
|          | C        | 0000         |
|          | PLKED    | 0004         |
|          | LKED     | 0000         |
| PH064S02 |          | 0000 or 0004 |
| PH064S03 |          | 0000         |

You can compare the output from this job to the sample output for DSNTEJ64, which is found in member DSN8TJ64 in data set named *prefix.SDSNIVPD*.

## Job DSNTEJ65

Job DSNTEJ65 demonstrates program preparation of an SQL procedure using the DB2 SQL procedures processor, DSNTPSMP. If you specified a remote location name on installation panel DSNTIPY, then jobs DSNTPSMP and DSNTIJSG, and the stored procedure job DSNTIJRX should be run on the remote server site. The major components of job DSNTEJ65 are:

- DSN8WLMP is a sample start-up procedure for a WLM-established stored procedures address space in which DSNTPSMP runs. DSN8WLMP is located in *prefix.SDSNSAMP*.
- DSN8ED4 is a sample C program that calls the DB2 SQL procedures processor.
- DSN8ES2 is a sample SQL procedure that calculates employee bonuses.
- DSN8ED5 is a sample C program that calls the SQL procedure DSN8ES2.

Before running DSNTEJ65, perform the manual editing described in the dependencies section of the job prolog. You must also manually tailor DSN8WLMP, the sample WLM startup procedure for DSNTPSMP. If DSNTEJ65 runs successfully, it produces the return codes shown in Table 97.

Table 97. DSNTEJ65 return codes

| Step     | PROCSTEP | Return code  |
|----------|----------|--------------|
| PH065S01 | PC       | 0000         |
|          | C        | 0000         |
|          | PLKED    | 0004         |
|          | LKED     | 0000         |
| PH065S02 |          | 0000 or 0004 |
| PH065S03 |          | 0000 or 0004 |
| PH065S04 | PC       | 0000         |
|          | C        | 0000         |
|          | PLKED    | 0004         |
|          | LKED     | 0000         |
| PH065S05 |          | 0000 or 0004 |
| PH065S06 |          | 0000         |

You can compare the output from this job to the sample output for DSNTEJ65, which is found in member DSN8TJ65 in the data set named *prefix.SDSNIVPD*.

## Starting an application in an ISPF/TSO environment in phase 6

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 247 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 sample applications panel, shown in Figure 44. Choosing option 3 on the sample applications panel during Phase 6

invokes the COBOL organization application, which uses DRDA access for distributed data. For more information, see “The distributed organization application scenario” on page 409.

---

## Phase 7: Accessing LOB Data

This optional phase demonstrates how to set up and use a DB2 LOB application. This phase creates an extension to the Employee sample database to manage employee resumes and photo images. You run these jobs in this phase:

- DSNTEJ7: Creates the Employee resume and photo table, then loads the resumes.
- DSNTEJ71: Populates the photo images, then validates that the resume and photo data is stored correctly.
- DSNTEJ73: Prepares an ISPF application for viewing employee resume data.
- DSNTEJ75: Prepares a GDDM application for viewing employee photo images.

Job DSNTEJ75 is not tailored by the installation CLIST unless you specify non-blank values for GDDM MACLIB and GDDM LOAD MODULES on panel DSNTIPW.

After you run these jobs, you can use ISPF and GDDM to view the sample employee resume and photo data.

### Job DSNTEJ7

Job DSNTEJ7 demonstrates how to create a LOB table with all the accompanying LOB table spaces, auxiliary tables, and indexes. It also demonstrates how to use the DB2 LOAD utility to load a CLOB column of fewer than 32 KB.

Job DSNTEJ7 consists of the following steps:

| Step | Function                                                |
|------|---------------------------------------------------------|
| 1    | Drops sample LOB objects                                |
| 2    | Creates sample Employee Resume and Photo LOB table      |
| 3    | Creates aliases for the table, then grants access to it |
| 4    | Uses the DB2 LOAD utility to load CLOB data             |
| 5    | Generates run-time statistics on the table              |

If DSNTEJ7 runs successfully, it produces the return code shown in Table 98.

*Table 98. DSNTEJ7 return codes*

| JOBSTEP | PROCSTEP | Return Code  |
|---------|----------|--------------|
| PH07S01 |          | 0000         |
| PH07S02 |          | 0000         |
| PH07S03 |          | 0000 or 0004 |
| PH07S04 | DSNUPROC | 0000         |
| PH07S05 | DSNUPROC | 0000         |

## Job DSNTSJ71

Job DSNTSJ71 compiles, link-edits, and binds two sample applications that manipulate LOB data. The DSN8DLPL sample application demonstrates how to use LOB locators to populate a LOB column larger than 32KB. The DSN8DLTC sample application validates the contents of the LOB table, verifying that it was populated correctly.

If DSNTSJ71 runs successfully, it produces the return codes shown in Table 99.

Table 99. DSNTSJ71 return codes

| JOBSTEP  | PROCSTEP | Return Code |
|----------|----------|-------------|
| PH071S01 | PC       | 0000        |
|          | C        | 0000        |
|          | PLKED    | 0004        |
|          | LKED     | 0000        |
| PH071S02 | PC       | 0000        |
|          | C        | 0000        |
|          | PLKED    | 0004        |
|          | LKED     | 0000        |
| PH071S03 |          | 0000        |
| PH071S04 |          | 0000        |
| PH071S05 |          | 0000        |

You can compare the output from this job with the sample output for DSNTSJ71 found in member DSN8TJ71 in your *prefix*.SDSNIVPD data set.

## Job DSNTSJ73

Job DSNTSJ73 compiles the DSN8DLRV sample application, which demonstrates how to use built-in functions like POSSTR and SUBSTR in order to traverse a CLOB column and break out data from it.

If DSNTSJ73 runs successfully, it produces the return codes shown in Table 100.

Table 100. DSNTSJ73 return codes

| JOBSTEP  | PROCSTEP | Return Code  |
|----------|----------|--------------|
| PH073S01 | PC       | 0004         |
|          | C        | 0000         |
|          | PLKED    | 0004         |
|          | LKED     | 0000         |
| PH073S02 | PC       | 0000         |
|          | C        | 0000         |
|          | PLKED    | 0004         |
|          | LKED     | 0000         |
| PH073S03 |          | 0000 or 0004 |

After running DSNTSJ73, you can view sample employee resumes by invoking DSN8DLRV as described in “Starting an application in an ISPF/TSO environment in phase 7” on page 395.

## Job DSNTEJ75

Job DSNTEJ75 runs sample program DSN8DLPV, which demonstrates how to manipulate BLOB data (employee photo images).

If DSNTEJ75 runs successfully, it produces the return codes shown in Table 101.

Table 101. DSNTEJ75 return codes

| JOBSTEP  | PROCSTEP | Return Code  |
|----------|----------|--------------|
| PH075S01 | PC       | 0000         |
|          | C        | 0000         |
|          | PLKED    | 0004         |
|          | LKED     | 0000         |
| PH075S02 |          | 0000 or 0004 |

After running DSNTEJ75, you can view sample employee photo images by invoking DSN8DLPV as described in “Starting an application in an ISPF/TSO environment in phase 7”.

## Starting an application in an ISPF/TSO environment in phase 7

You must have access to ISPF load module libraries in order to run the Employee Resume and Photo sample applications. See “Making panels, messages, and load modules available to ISPF and TSO” on page 247 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 Employee Resume and Photo sample applications, enter:

```
CALL 'prefix.RUNLIB.LOAD(DSN8SDM)'
```

After you enter this command, DB2 displays the sample applications panel, shown in Figure 44.

Choosing option 4 on the sample applications panel during Phase 7 invokes the “Employee Resume” sample application, which processes CLOB data. Choosing option 5 on the sample applications panel during Phase 7 invokes the “Employee Photo” sample application, which processes BLOB data. For more information, see “Employee resume and photo scenarios” on page 417.

---

## The sample applications

This section describes the sample applications. The names of the sample applications have changed for Version 7. Check to make sure you have the authority to run the Version 7 sample programs. For information on granting and revoking DB2 privileges, see Part 3 (Volume 1) of *DB2 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 your *prefix.SDSNIVPD* data set.

## 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, IEBTPCH, 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 |
|---------------------|---------------------------------|
|---------------------|---------------------------------|

- |                |            |
|----------------|------------|
| • MAJOR SYSTEM | • ACTION   |
| • ACTION       | • OBJECT   |
| • OBJECT       | • SEARCH   |
| • SEARCH       | CRITERIA   |
| CRITERIA       | • LOCATION |
| • DATA         | • 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 13. Connecting distributed database systems” on page 465.

**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:

- |           |                                                                                |
|-----------|--------------------------------------------------------------------------------|
| <b>DE</b> | Department—general department and manager information for department specified |
| <b>DS</b> | Department structure—hierarchy information for department specified            |
| <b>EM</b> | 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 102 demonstrates three ways to use these characters to create a DATA value.

*Table 102. 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 53 on page 402, 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 five scenarios for using the sample applications:

- “The project application scenario”
- “The organization application scenario” on page 401
- “The phone application scenario” on page 405
- “The distributed organization application scenario” on page 409
- “Employee resume and photo scenarios” on page 417

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 373 and 392
- “Starting an application in an IMS environment” on page 376
- “Starting an application in a CICS environment” on page 380.

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 50. 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 51 on page 401.

```

                                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 51. 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 52 on page 402.

```

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 52. 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 53. 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 53. 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 54. 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 54. 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”, 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 55.

```

      ERASING A DEPARTMENT
MAJOR SYSTEM ...: 0      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 55. 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 403, 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 56 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                          |
| NAME           | : SPIFFY COMPUTER SERVICE DIV. |
| MANAGER ID     | : 000010                       |
| ADMIN DEP ID   | : A00                          |
| EMPLOYEE ID    | : 000050                       |
| FIRST NAME     | : JOHN                         |
| MIDDLE INITIAL | : B                            |
| LAST NAME      | : GEYER                        |
| WORK DEPT ID   | : A00                          |

PFK: 02=RESEND 03=END

*Figure 56. 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 (DSN8710.EMP) and the department table (DSN8710.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 103 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 44 on page 373.

Table 103. 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 57.

----- 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 57. 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 58 shows the updated panel.

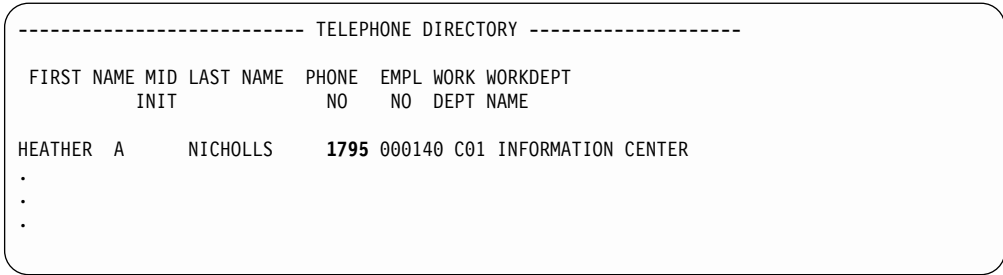


Figure 58. Telephone application—updated display

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 104.

Table 104. 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 59 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 59. 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.

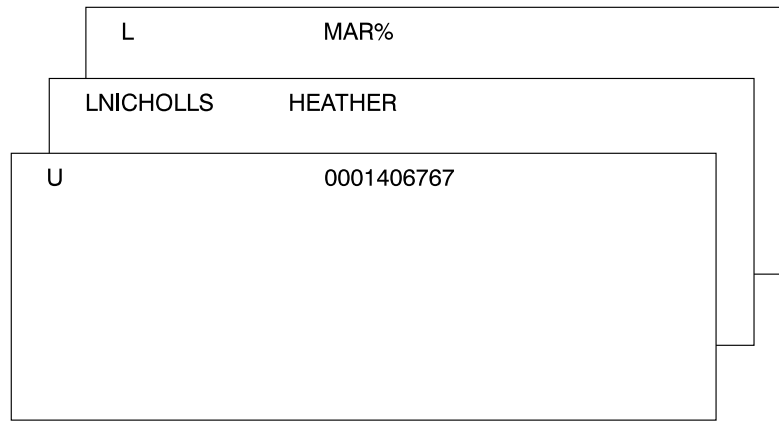


Figure 59. Example of a card image data set

The other data set that contains the JCL is supplied with DB2 and is contained in DSNTEJ2P, which is part of *prefix*.SDSNSAMP. Figure 60 shows the data set that contains the JCL with the card image data sets embedded.

```
//PH02PS05 EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//REPORT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CARDIN DD *
U                                0001406767
LNICHOLLS          HEATHER
L                  MAR%
//SYSTSIN DD *
DSN SYSTEM(DSN)
RUN  PROGRAM(DSN8BP3) PLAN(DSN8BP71) LIB('prefix.RUNLIB.LOAD')
END
```

Figure 60. The job control language data set

The complete data set can be submitted to the system either through a card reader or from a terminal through TSO. Figure 61 on page 409 is an example of the batch output.

| ----- TELEPHONE DIRECTORY ----- |            |         |                 |                    |              |                        |
|---------------------------------|------------|---------|-----------------|--------------------|--------------|------------------------|
| LAST NAME                       | FIRST NAME | INITIAL | PHONE<br>NUMBER | EMPLOYEE<br>NUMBER | WORK<br>DEPT | WORK<br>DEPT NAME      |
| QUINTANA                        | DOLORES    | M       | 6767            | 000130             | C01          | INFORMATION CENTER     |
| NICHOLLS                        | HEATHER    | A       | 1793            | 000140             | C01          | INFORMATION CENTER     |
| SCOUTTEN                        | MARILYN    | S       | 1682            | 000180             | D11          | MANUFACTURING SYSTEMS  |
| PEREZ                           | MARIA      | L       | 9001            | 000270             | D21          | ADMINISTRATIVE SYSTEMS |

Figure 61. Example of the phone application batch output

## 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 62. 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 63. 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 64. 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 65. 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 66. 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 67. 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 68. 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 69. 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
===>_
DSN80141 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 70. 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 71. 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 72. 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 73. 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 74. 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 75. 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 76. 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

==>_
DSN80031  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 77. Employee at a remote location erased

## Employee resume and photo scenarios

The LOB sample application extends the existing DB2 sample employee database by adding a new table for storing employee resumes and photographs as CLOB and BLOB entries. The supporting JCL, application objects, and input data for this table are provided.

The purpose of the LOB sample application is to:

- Provide an IVP for LOB functions
- Demonstrate how to:
  - Use DDL to create LOB objects

- Use the DB2 LOAD utility to populate LOB columns of 32K bytes or less
- Create an application program to populate LOB columns of greater than 32K bytes
- Use LOB locators and related functions to manipulate LOBs without materializing the data.

The LOB sample application consists of a batch portion and an optional online portion. The batch portion verifies that LOB objects can be created, populated, and read successfully using locators and supporting functions. The online portion demonstrates further techniques for using and manipulating LOB data.

The batch portion consists of four jobs: DSNTEJ7, DSNTEJ71, DSNTEJ73, and DSNTEJ75. For more information about these batch jobs see “Phase 7: Accessing LOB Data” on page 393.

The optional online portion of the LOB sample application adds two additional scenarios to the existing three scenarios provided in IVP phase 3. These scenarios include the following activities by the user:

1. Invoking the DB2 sample connection manager to display the DB2 ISPF sample application menu, DSN8SSM
2. Selecting and viewing sample employee resumes by using option 4 of DSN8SSM
3. Selecting and viewing sample employee photo images by using option 5 of DSN8SSM

All application programs for the LOB sample are written in C language. The resume viewer requires ISPF. The photo viewer requires ISPF and GDMM.

### Sample LOB table

The LOB sample application uses the EMP\_PHOTO\_RESUME table. The sample jobs create, load and manipulate the table.

Table 105. EMP\_PHOTO\_RESUME table

| Column Name: | EMPNO               | EMP_ROWID      | PSEG_PHOTO            | BMP_PHOTO            | RESUME              |
|--------------|---------------------|----------------|-----------------------|----------------------|---------------------|
| Type:        | CHAR(6)<br>NOT NULL | ROWID          | BLOB(500K)            | BLOB(100K)           | CLOB(5K)            |
| Description: | Employee number     | Row identifier | Employee photo (PSEG) | Employee photo (BMP) | Employee resume     |
| Values:      | 000130              | #####          | Delores M. Quintana   | Delores M. Quintana  | Delores M. Quintana |
| Values:      | 000140              | #####          | Heather A Nicholls    | Heather A Nicholls   | Heather A Nicholls  |
| Values:      | 000150              | #####          | Bruce Adamson         | Bruce Adamson        | Bruce Adamson       |
| Values:      | 000190              | #####          | James H. Walker       | James H. Walker      | James H. Walker     |

### LOB application panels for Resume scenario

The LOB application begins with the panel shown in Figure 44 on page 373. Select option 4 from the DB2 Sample Application Menu, DSN8SSM, to look at the resumes

under ISPF. DSN8DLRV shows the following panels when selecting and viewing sample employee resumes until the user signals ISPF to exit by pressing the END key.

Prompt the user to select from the list of available resumes by displaying ISPF panel DSN8SSE:

```

DSN8SSE                                DB2 EMPLOYEE SELECTION PANEL
===>_

SELECT ONE OF THE EMPLOYEES AND PRESS ENTER.

1. 000130 - DELORES M. QUINTANA
2. 000130 - HEATHER A. NICHOLLS
3. 000130 - BRUCE ADAMSON
4. 000130 - JAMES H. WALKER

PRESS:      END TO EXIT

```

*Figure 78. DB2 Employee Selection Panel*

The employee serial numbers and names are hard-coded into the panel because the data for this sample is predetermined. Display the formatted resume on ISPF panel DSN8SSR:

```

DSN8SSR                                DB2 EMPLOYEE RESUME APPLICATION
===>_

PERSONAL INFORMATION:                  DEPARTMENT INFORMATION:
- NAME: DELORES M. QUINTANA            - EMPLOYEE NO. : 000130
- HOME: 1150 EGLINTON AVE              - DEPARTMENT NO.: C01
      MELLONVILLE, IDAHO 83757        - MANAGER      : SALLY KWAN
      (208) 555-9933                    - POSITION      : ANALYST
- BORN: SEPTEMBER 15, 1925             - PHONE       : (208) 555-4578
- SEX: FEMALE HT:5'2" WT:120 LBS.      - HIRE DATE    : 1971-07-28
- MARITAL STATUS: MARRIED

EDUCATION:
1965 MATH AND ENGLISH B.A.            1960 DENTAL TECHNICIAN
      ADELPHI UNIVERSITY                  FLORIDA INSTITUTE OF TECHNOLOGY

WORK HISTORY:
10/91 - PRESENT ADVISORY SYSTEMS ANALYST
      PRODUCING DOCUMENTATION TOOLS FOR ENGINEERING DEPARTMENT
12/85 - 9/91   TECHNICAL WRITER
      WRITER, TEXT PROGRAMMER, AND PLANNER
1/79 - 11/85   COBOL PAYROLL PROGRAMMER
      WRITING PAYROLL PROGRAMS FOR A DIESEL FUEL COMPANY

```

*Figure 79. DB2 Employee Resume Application*

This panel is designed around the sample data. It is assumed that all resume information for an employee will fit predictably and no handling for special cases is provided. The "Interests" section of the resume is not presented due to space constraints on the panel.

DSN8DLRV is written in C language and linked with ISPF and DB2 Call Attach Facility. The package name and plan name are both DSN8LRvr, where vr is the DB2 version and release.

### LOB application panels for Photo scenario

The LOB application begins with the panel shown in Figure 44 on page 373. Select option 5 from the DB2 Sample Application Menu, DSN8SSM, to look at employee photos using GDDM. This application requires that you include the GDDM load module library (SADMMOD) in your logon procedure or in the ISPLLIB concatenation.

DSN8DLRV shows the following panels when selecting and viewing sample employee photos. Prompt the user to select from the list of available photos by displaying ISPF panel DSN8SSE:

```
DSN8SSE                                DB2 EMPLOYEE SELECTION PANEL
===> _

SELECT ONE OF THE EMPLOYEES AND PRESS ENTER.

1. 000130 - DELORES M. QUINTANA
2. 000130 - HEATHER A. NICHOLLS
3. 000130 - BRUCE ADAMSON
4. 000130 - JAMES H. WALKER

PRESS:      END TO EXIT
```

Figure 80. DB2 Employee Selection Panel

Select END to exit.

DSN8DLPV is a C language program linked with ISPF, GDDM and the DB2 Call Attach Facility. The package name and plan name are both DSN8LPvr, where vr is the DB2 version and release. To run DSN8DLPV you must include the GDDM load module library (SADMMOD) in the logon procedure or in the ISPLLIB concatenation.

---

## Edit exit routine

The edit exit routine is prepared in Phase 1. It works with the employee table (DSN8710.EMP) and is written in assembler language.

The name of the edit exit routine is DSN8EAE1. When the employee table (DSN8710.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. DSNTESA is used in Phase 3 of the verification process.

The first group of statements in DSNTESA create a temporary work file table space and defines a created temporary 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 DSN8710.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 = 'DSN8610'
 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 DSNTPE2. If SPUFI is not bound when you want to execute these queries, you can use the Version 6 DSNTPE2.

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.

---

## Dynamic SQL programs (DSNTIAD, DSNTPE2, DSNTIAUL)

SPUFI is a part of the distributed product. An installation 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. These programs are documented in an appendix of *DB2 Utility Guide and Reference* and *DB2 Application Programming and SQL Guide*



---

## Chapter 11. 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 Part 3 (Volume 1) of *DB2 Administration Guide*.

---

### Making DB2 load modules available to IMS

If you have already included the *prefix.SDSNLOAD* library in your LNKLISTxx, 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 LNKLISTxx, 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 Part 4 of *DB2 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 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 429.

**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 429 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 429.

#### **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 106. 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 Part 4 of *DB2 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 107 on page 429 shows the different possibilities of SSM specifications.

Table 107. 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 10. Verifying with the sample applications” on page 349.

## 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. The recommendation is 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 108.

*Table 108. 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 431.

## 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 109.

*Table 109. 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.

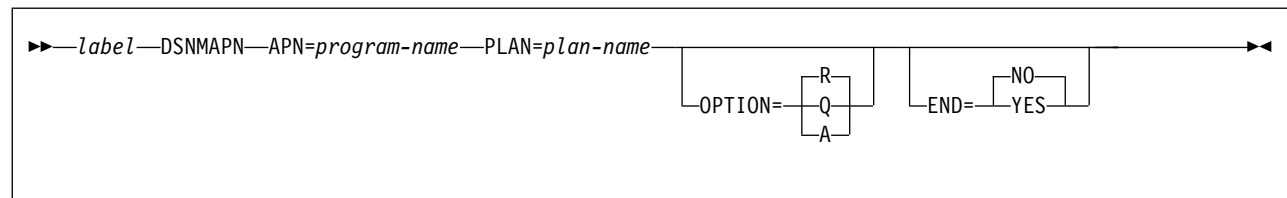


### IMS attachment facility macro (DSNMAPN)

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 429.



**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=RIQIA**

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=NOIYES**

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 12. 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. If you are running CICS Transaction Server (CICS TS) Version 1.2 or higher, see the *CICS DB2 Guide* for installation instructions.

To connect the CICS attachment facility, whether installing or migrating, you must do the following:

1. Calculate space requirements for the CICS attachment facility. Recalculate if you are migrating.
2. Define the DB2 to CICS connection using the RCT. For migration either to CICS Version 4 or to CICS TS Version 1.1, reassemble your RCT.
3. Update the CICS system tables. (Not always necessary for migration.)
4. Update the CICS initialization JCL.
5. Coordinate DB2 and CICS security. (Not always necessary for migration.)
6. 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 Part 3 (Volume 1) of *DB2 Administration Guide*.

---

### Using the attachment facility for CICS Version 4 and CICS TS Version 1.1

The CICS-DB2 attachment facility is provided on the CICS product tape. Each CICS region connecting to DB2 should use the attachment facility provided on the CICS product tape.

CICS Version 4 and subsequent releases 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 the *CICS for MVS/ESA Installation Guide* before deleting program definitions. Check for down-level attachment facility program definitions in your CSD. Delete any of these program definitions:

|          |          |          |          |
|----------|----------|----------|----------|
| DSNCCOM0 | DSNCCOM1 | DSNCCOM2 | DSNCEDF1 |
| DSNCEDON | DSNCEXT1 | DSNCEXT2 | DSNCMSG0 |
| DSNCSTOP | DSNCSTRT | DSNCUEXT |          |

Delete any of these transaction definitions if:

- DSNC
- any transaction that executes program DSNCCOM0, DSNCCOM1, or DSNCCOM2

Most module names for the attachment facility have been renamed DSN2xxxx.

---

### 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.

For storage estimates for each CICS region, see Table 112 on page 435, and the *CICS DB2 Guide*. For more information on CICS storage considerations, see the appropriate CICS performance guide for your release.

## 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.

Table 110. Virtual storage requested for CICS region

| 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 435 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.<br><br>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.<br><br>Above 16M.                 |
| "                | 0.8KB  | CICS Control Block                  | Per active thread  | Above 16M.                                                                                                                                                   |
| "                | 0.6KB  | CICS GETMAINed Storage              | Per address space  | Above 16M.                                                                                                                                                   |

Table 111. Contents of CICS address space high private area

| Storage Area         | Amount | Description                | Factor            | Notes |
|----------------------|--------|----------------------------|-------------------|-------|
| LSQA, SP229, and 230 | 5.5KB  | MVS and DB2 Control Blocks | Per active thread |       |

Table 111. Contents of CICS address space high private area (continued)

| Storage Area                               | Amount | Description                                    | Factor            | Notes                                                                                                                                            |
|--------------------------------------------|--------|------------------------------------------------|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| MVS Free Storage<br>(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. |

Table 112. CICS storage summary

| 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         |
| <b>Total</b>                            | <b>43.8</b>            | <b>12.5KB</b> |

## 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 113 shows their default values.)

Table 113. Default COMD and POOL entry 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 a complete description of this macro and how to assemble it, see “CICS attachment facility macro (DSNCRCT)” on page 444. For information on specifying the RCT suffix, see page 442.

Plan for performance before connecting DB2 to CICS because some of the parameters influence performance. See Part 5 (Volume 2) of *DB2 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 377.

---

## 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.

### System initialization table entries

- Program list table (PLT) processing (optional)

When using PLT processing, enter the PLT suffixes to the PLTP1 entry, the PLTSD entry, or both. For an explanation of PLT processing, see “PLT processing” on page 440.

- CICS tracing (optional)

DB2 uses the CICS trace and auxiliary trace facilities for tracing. See *DB2 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 DSNF 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. The CICS-defined group named DFHDB2 that contains entries for the CICS attachment facility

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 433 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 442 and the GROUP entry on page 455.

- 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

The attachment facility definitions are supplied automatically. You can use RDO to define sample applications and optional entries.

### **Resource Definition Online (RDO):**

- 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

Do not create a transaction entry for transaction DSNC, which is automatically supplied by CICS. You can optionally define one or more DSNC subcommands, or the first four characters of DB2 commands, as transactions that execute DSN2COM1.

For more information on DSNC commands, see “Providing CICS support for DB2 commands” on page 441.

- DB2 sample applications

Phase 5 of installation verification requires that program, mapset, and transaction resources be defined to CICS. See Table 114 for a list of required entries. For more information on the verification procedure, see “Chapter 10. Verifying with the sample applications” on page 349.

- User transactions

User transactions accessing DB2 resources must also be defined to CICS.

Table 114 shows examples of transaction definitions.

*Table 114. CICS transaction resource definitions for DB2*

| Functions Defined to CICS     | Resource Definition                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DB2 Commands entered directly | <p>As many as wanted, for example:</p> <pre> DEFINE TRANSACTION(DISC) PROGRAM(DSN2COM1) GROUP(DB2-group) TWSIZE(1200)  DEFINE TRANSACTION(DISP) PROGRAM(DSN2COM1) GROUP(DB2-group) TWSIZE(1200)  DEFINE TRANSACTION(MODI) PROGRAM(DSN2COM1) GROUP(DB2-group) TWSIZE(1200)  DEFINE TRANSACTION(STOP) PROGRAM(DSN2COM1) GROUP(DB2-group) TWSIZE(1200)  DEFINE TRANSACTION(STRT) PROGRAM(DSN2COM1) GROUP(DB2-group) TWSIZE(1200)  DEFINE TRANSACTION(-DIS) PROGRAM(DSN2COM1) GROUP(DB2-group) TWSIZE(1200)  (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>DB2 Command Reference</i> for all the valid commands. See also “Providing CICS support for DB2 commands” on page 441 for more information.) </pre> |
| User transactions             | <p>As many as needed:</p> <pre> DEFINE TRANSACTION(TRN1) PROGRAM(PROGRAM1) GROUP(user-group) </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |



Table 114. CICS transaction resource definitions for DB2 (continued)

| 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:**

1. To make the resource definitions available to CICS, issue the following commands:
 

```

CEDA INSTALL GROUP(DB2-group)
CEDA INSTALL GROUP(user)

```
2. Do not create entries for DSNB commands. These are automatically defined by the attachment facility.
3. Transaction definitions for PROGRAM(DSN2COM1) should include the TASKDATALOC(ANY) parameter, as in the following example:
 

```

DEFINE TRANSACTION(DISC) PROGRAM(DSN2COM1)
GROUP (GROUP) TWASIZE(1200)
TASKDATALOC(ANY) TASKDATAKEY(CICS)

```
4. 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.
5. 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.

## Program entries

Ensure that you use the CICS-supplied resource group DFHRMI (included in the DFHLIST list) to specify the resource manager interface (RMI).

Table 115. CICS program and map resource definitions for DB2

| Functions Defined to CICS | Resource Definition                                                                                         |
|---------------------------|-------------------------------------------------------------------------------------------------------------|
| User transactions         | <p>As many as needed:</p> <pre> DEFINE PROGRAM(program-name)   GROUP(user-group) LANGUAGE(prog-lang) </pre> |

Table 115. CICS program and map resource definitions for DB2 (continued)

| Functions Defined to CICS                                   | Resource Definition                                         |
|-------------------------------------------------------------|-------------------------------------------------------------|
| Sample application transactions during install verification | DEFINE PROGRAM(DSN8CP0)<br>GROUP(DB2-group) LANGUAGE(PLI)   |
|                                                             | DEFINE PROGRAM(DSN8CP1)<br>GROUP(DB2-group) LANGUAGE(PLI)   |
|                                                             | DEFINE PROGRAM(DSN8CP2)<br>GROUP(DB2-group) LANGUAGE(PLI)   |
|                                                             | DEFINE PROGRAM(DSN8CP3)<br>GROUP(DB2-group) LANGUAGE(PLI)   |
|                                                             | DEFINE PROGRAM(DSN8CP6)<br>GROUP(DB2-group) LANGUAGE(PLI)   |
|                                                             | DEFINE PROGRAM(DSN8CP7)<br>GROUP(DB2-group) LANGUAGE(PLI)   |
|                                                             | DEFINE PROGRAM(DSN8CP8)<br>GROUP(DB2-group) LANGUAGE(PLI)   |
|                                                             | DEFINE PROGRAM(DSN8CC0)<br>GROUP(DB2-group) LANGUAGE(COBOL) |
|                                                             | DEFINE PROGRAM(DSN8CC1)<br>GROUP(DB2-group) LANGUAGE(COBOL) |
|                                                             | DEFINE PROGRAM(DSN8CC2)<br>GROUP(DB2-group) LANGUAGE(COBOL) |
|                                                             | DEFINE MAPSET(DSN8CPG)<br>GROUP(DB2-group)                  |
|                                                             | DEFINE MAPSET(DSN8CPD)<br>GROUP(DB2-group)                  |
|                                                             | DEFINE MAPSET(DSN8CPF)<br>GROUP(DB2-group)                  |
|                                                             | DEFINE MAPSET(DSN8CPE)<br>GROUP(DB2-group)                  |
|                                                             | DEFINE MAPSET(DSN8CCG)<br>GROUP(DB2-group)                  |
|                                                             | DEFINE MAPSET(DSN8CCD)<br>GROUP(DB2-group)                  |
|                                                             | DEFINE MAPSET(DSN8CPN)<br>GROUP(DB2-group)                  |
|                                                             | DEFINE MAPSET(DSN8CPL)<br>GROUP(DB2-group)                  |
|                                                             | DEFINE MAPSET(DSN8CPU)<br>GROUP(DB2-group)                  |

**Note:**

1. To make the resource definitions available to CICS, issue the following install commands:  
CEDA INSTALL GROUP(DB2-group)  
CEDA INSTALL GROUP(user-group)
2. 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 116 on page 441 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.

## 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=DSN2COM0.
- 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=DSN2COM2.
- 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 437 and “Program entries” on page 439 for more information on required entries for PLT processing.

For a list of optional entries on the PLT, refer to Table 116.

*Table 116. Optional CICS program list table (PLT) entries for DB2*

| Functions Defined to CICS              | PLT Entry                                    |
|----------------------------------------|----------------------------------------------|
| Post-initialization attachment program | DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM<br>:<br>: |
|                                        | DFHPLT TYPE=ENTRY,PROGRAM=DSN2COM0           |
| First quiesce stage shutdown program   | DFHPLT TYPE=ENTRY,PROGRAM=DSN2COM2<br>:<br>: |
|                                        | DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM           |

**Note:**

1. DSN2COM0 must be defined in the last stage of CICS initialization (PLTPI).
2. DSN2COM2 must be defined in the first quiesce stage of CICS shutdown (PLTSD). To ensure that DSN2COM2 executes during the first quiesce stage, be sure that the DSN2COM2 program entry is defined before the DFHDELIM program entry.

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, DSN2COM1.

DSN2COM1 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 114 on page 438 allows a user to enter DISP as a transaction ID. DSN2COM1 recognizes this request as equivalent to DSNCRCT DISP. In addition, it is also possible to create transaction entries for DSN2COM1 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 DSNCRCT subcommands are documented in *DB2 Command Reference*.

To provide support for command processing, you must:

- Define a program resource to CICS using RDO for the command processor, DSN2COM1.
- 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 DSN2COM1 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 *DB2 Command Reference*.

---

## Updating CICS initialization JCL

**DB2 libraries:** You need to specify the DB2 program library (SDSNLOAD)library and the RCT library in the STEPLIB/JOBLIB concatenation in your CICS startup job or in the LINKLSTxx.

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 433.

**RCT Suffix:** The attachment facility allows multiple RCTs to coexist by using a suffix field for identification. The default RCT suffix is 00. For more information on using the RCT suffix, see page 449.

---

## 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)  
CICS can use a default USERID if you are not signed on.
- The CICS sign-on operator ID (OPIDENT)

Specify the OPIDENT using the CICS option of the RACF ADDUSER command. For more information, see *OS/390 Security Server (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. 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 Part 3 (Volume 1) of *DB2 Administration Guide*.

The ability to enter DB2 and attachment commands from CICS terminals can also be controlled through standard CICS security mechanisms. See Part 4 (Volume 1) of *DB2 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 CICS-supplied SQL language interface module 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 INQUIRE EXITPROGRAM.

INQUIRE EXITPROGRAM 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. 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 381.

For more information about preparing CICS applications for DB2, see Part 4 of *DB2 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).

This macro supports DSNTJE5A, which optionally assembles and links a RCT. See “Job DSNTJE5A” on page 377 for more information. If you use the CICS-supplied procedures, you must make the following changes:

- 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.

The RCT must be reassembled when you migrate from one release of the attachment facility to another.

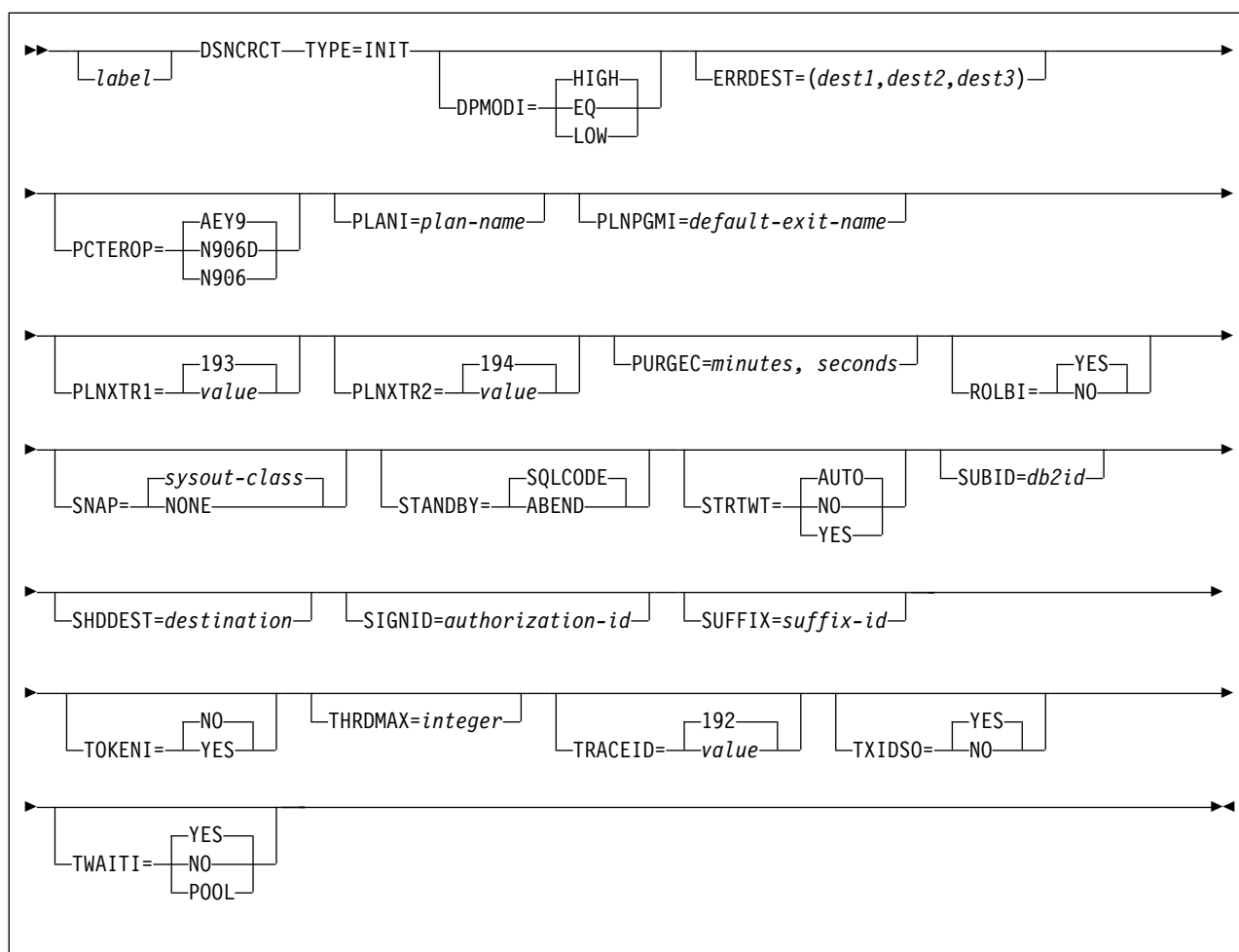
The TYPE=INIT macro must be specified first. If the TYPE=COMD macro is specified, it must be second. If TYPE=POOL is specified, it must be after TYPE=INIT and TYPE=COMD (if there is a COMD) and before TYPE=ENTRY. The TYPE=ENTRY macro can be specified an indefinite number of times to identify specific 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.
- 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*, specify 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 DSN 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=AEY9IN906DIN906**

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 DSN transaction dump is taken, and whether the DSNCSQL 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 DSN transaction dump is to be taken and the DSNCSQL RMI associated with the transaction is to be disabled. Disabling the DSNCSQL 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 DSN transaction dump is to be taken and the DSNCSQL 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 DSN transaction dump is **not** to be taken and the DSNCSQL 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.

**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:** DSNCEXT

**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*, specify a number from 256 to 511.

**Default:** 449

**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*, specify a number from 256 to 511.

**Default:** 450

**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.

**Default:** (0,30)

**ROLBI=YESNO**

Specifies a default for the ROLBE parameter in other TYPEs of the DSNCRCT 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 DSNCRCT DISP STAT command) during shutdown of the attachment facility. For *destination*, specify 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*, specify 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 “DSNCRCT TYPE=ENTRY” on page 453. 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*, specify 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*NOIYES**

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 is complete. However, the attachment facility can be terminated by the DSNCRCT 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 DSNCRCT 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*, specify 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 two characters for CICS Version 4 and subsequent releases. 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 *DB2 Command Reference*.

You can use the INITPARM command to select the RCF suffix. For example:

```
INITPARM=(DSNCRCT='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:**00

**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 138.

**Default:** 12

**TOKENI=NOIYES**

Specifies the default TOKENE if TOKENE is not specified on the TYPE=ENTRY statement. For more information on TOKENE, see page 459.

**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 256 through 511, that is valid to CICS and that does not conflict with values already in use for CICS trace can be specified.

**Default:** 451

**TXIDSO=YESINO**

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.

**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

**TWAITI=YESINOIPOOL**

Specifies the default value (YES, NO, or POOL) that is to be created for the TWAIT parameter on other types of the macro.

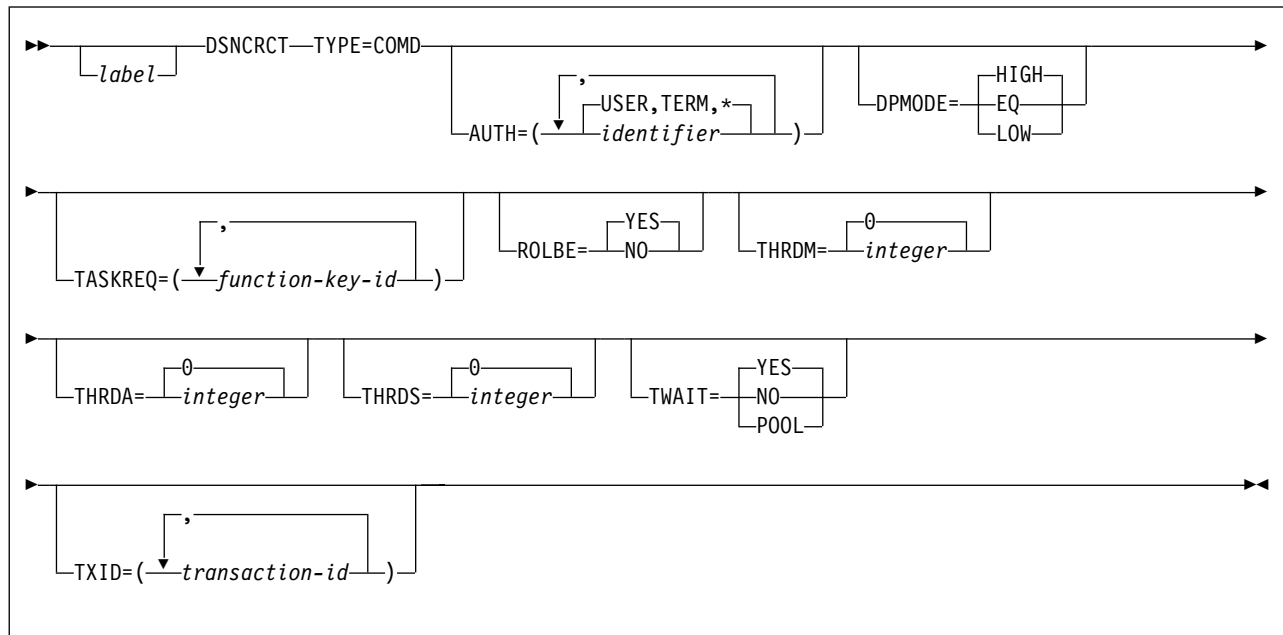
**Default:** YES

For a description of the TWAIT parameter, refer to “DSNCRCT TYPE=ENTRY” on page 453.

## 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 454.

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.

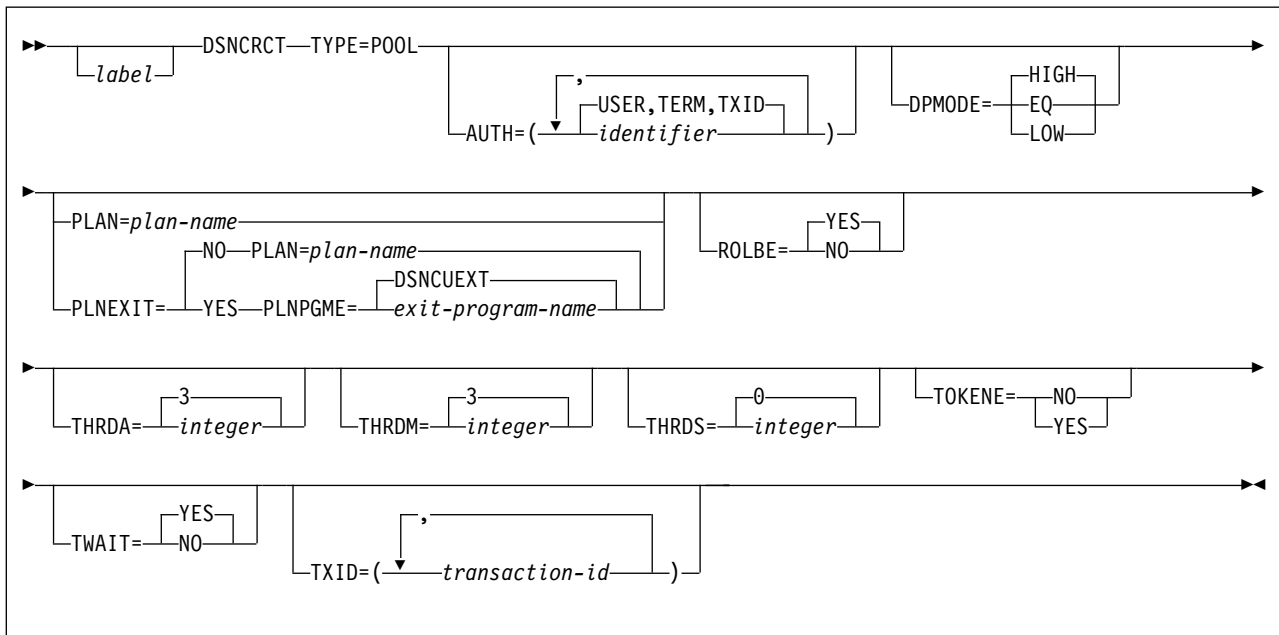
It is necessary only to specify DSNCRCT (or your user-defined command thread transaction) for the TXID parameter, even if you are using multiple transaction IDs to execute DSN2COM1.

For more information on the TYPE=COMD parameters, see the descriptions under TYPE=ENTRY on page 453.

## 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”.

#### Usage notes:

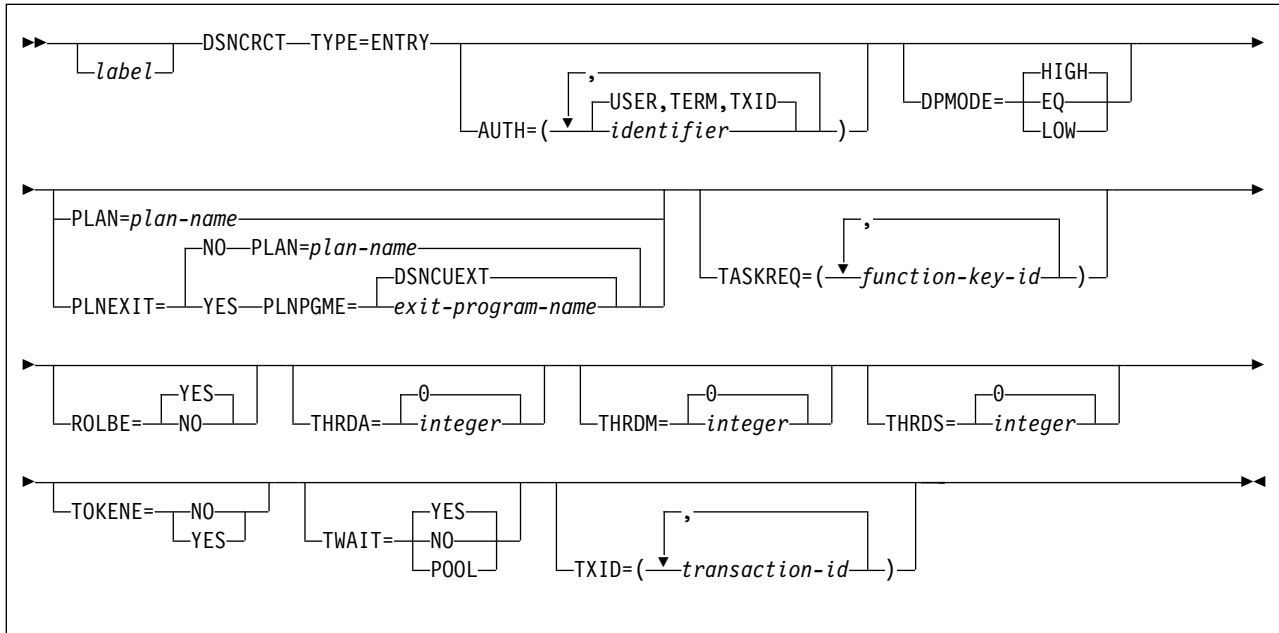
- When the pool is used to create a thread for a transaction, the thread is terminated when the transaction reaches SYNCPOINT (terminal-driven transactions) or EOT (non-terminal-driven transactions), unless another transaction is queued for the thread with the same plan.
- The pool provides default processing 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 *DB2 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*, specify either an explicit character string or specify 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*, specify 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

CICS sign-on user ID (USERID)

RACF connected group name

### How DB2 interprets values

Represents the primary DB2 authorization ID.

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 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,\*,\*)

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,\*,\*)

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 *DB2 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*, specify 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 460.

**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 *DB2 Administration Guide*.

**Default:** DSNCUEXT

**ROLBE=YESNO**

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*, specify a value that does not exceed 99 or the THRDM value. The general restriction is  $(THRDS \leq THRDA \leq THRDM \leq 99)$ . 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 *DB2 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*, specify an integer value that does not exceed 99. The general restriction is  $(THRDS \leq THRDA \leq THRDM \leq 99)$ .

The value specified for THRDM is the maximum value that can be specified in the DSNB MODIFY TRANSACTION command when using that command to change the value of THRDA. See Part 4 (Volume 1) of *DB2 Administration Guide* for a description of the DSNB 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 *DB2 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*, specify a value that does not exceed the THRDA value or 99. The general restriction is  $(THRDS \leq THRDA \leq THRDM \leq 99)$ .

For considerations using threads in conjunction with dynamic plan selection, see Appendixes (Volume 2) of *DB2 Administration Guide*.

**Default:** 0

**TOKENE=NOIYES**

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 Part 5 (Volume 2) of *DB2 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=YESINOIPOOL**

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*, specify 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
```

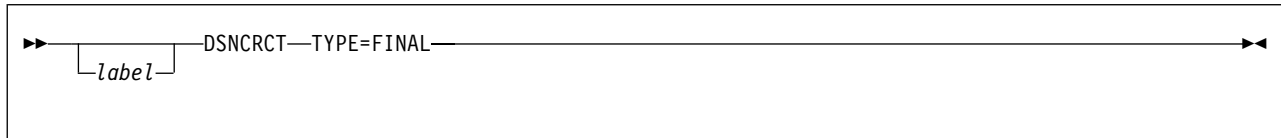
3. If you want separate CICS attachment facility statistics for each transaction, you must code a separate entry for each transaction, as in this example:

```
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.



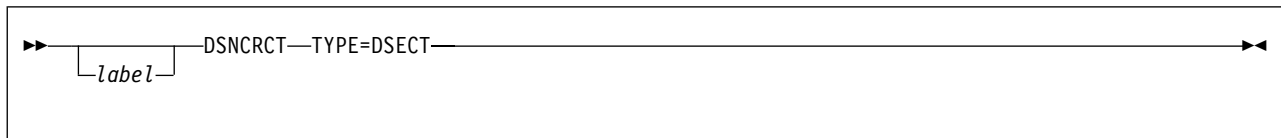
### Description

#### 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.





---

## Part 3. Communicating with other systems

|                                                            |     |
|------------------------------------------------------------|-----|
| <b>Chapter 13. Connecting distributed database systems</b> | 465 |
| The database protocols (DRDA vs private)                   | 465 |
| The communications protocols.                              | 465 |
| The role of the communications database (CDB)              | 466 |
| Enhancements since Version 5                               | 467 |
| DB2 installation considerations                            | 467 |
| <br><b>Chapter 14. Connecting systems with VTAM</b>        | 469 |
| Step 1: Customize VTAM for DB2                             | 471 |
| Step 2: Choose names and a password                        | 471 |
| Names for the local subsystem                              | 471 |
| A password for the local subsystem.                        | 472 |
| Names you need from the remote systems                     | 472 |
| Names chosen by Spiffy Computer Company                    | 473 |
| Step 3: Define the DB2 subsystem to VTAM                   | 473 |
| The APPL statement                                         | 473 |
| Options for which you must choose values                   | 474 |
| Options you must code exactly as given                     | 476 |
| Options that must use VTAM defaults                        | 476 |
| Other options of interest                                  | 477 |
| Options ignored by DB2                                     | 477 |
| The modeent macro                                          | 477 |
| Default modes.                                             | 478 |
| Sample mode entries                                        | 478 |
| Modeent options                                            | 479 |
| Step 4: Populate the communications database               | 480 |
| SYSIBM.LOCATIONS table                                     | 480 |
| SYSIBM.LUNAMES table                                       | 481 |
| SYSIBM.USERNAMES table                                     | 483 |
| Step 5: Start VTAM to use DB2                              | 483 |
| Step 6: Tune the system                                    | 484 |
| Controlling buffer storage                                 | 485 |
| Increase the number of IOBUF buffers.                      | 485 |
| Decrease the session level pacing count                    | 485 |
| Decrease the number of concurrent conversations            | 485 |
| Decrease the request unit (RU) size                        | 486 |
| Controlling pacing                                         | 486 |
| Recommendation for APPL pacing option.                     | 486 |
| Recommendation for MODEENT pacing options                  | 487 |
| Modifying default session limits                           | 487 |
| Increasing session limits for specific modes and lus       | 487 |
| Increasing session limits for all modes and LUs            | 488 |
| Modifying class of service                                 | 488 |
| Associating applications with modes                        | 488 |
| Update LUNAMES to associate modes with LU names.           | 488 |
| Update SYSIBM.LUMODES with conversation limits.            | 489 |
| Update SYSIBM.MODESELECT to associate plans with modes     | 490 |
| Updating CDB values                                        | 491 |
| Calculating session limits.                                | 492 |
| Calculating session limits for DB2 private protocol access | 492 |
| Considerations for mixed applications                      | 493 |
| Calculating VTAM I/O buffer pool (IOBUF) storage           | 495 |
| Interpreting CNOS messages                                 | 496 |

|                                                                     |            |
|---------------------------------------------------------------------|------------|
| Sample VTAM definitions to connect two DB2s . . . . .               | 499        |
| Basic definitions . . . . .                                         | 499        |
| Channel-connected DB2s . . . . .                                    | 502        |
| NCP-connected DB2s . . . . .                                        | 503        |
| Using the change log inventory utility to update the BSDS . . . . . | 507        |
| <br><b>Chapter 15. Connecting systems with TCP/IP . . . . .</b>     | <b>509</b> |
| Enabling TCP/IP communication . . . . .                             | 510        |
| TCP/IP limitations . . . . .                                        | 510        |
| Initializing a TCP stack for use with a VIPA . . . . .              | 511        |
| Using two-phase commit . . . . .                                    | 511        |
| Multiple DB2 subsystems on a single OS/390 image . . . . .          | 511        |
| Multiple DB2 subsystems with multiple TCP/IP stacks . . . . .       | 512        |
| Multiple DB2 subsystems with one TCP/IP stack . . . . .             | 512        |
| Step 1: Prepare the Language Environment runtime library . . . . .  | 512        |
| Step 2: Enable DDF for UNIX System Services . . . . .               | 513        |
| Step 3: Define the DB2 subsystem to TCP/IP . . . . .                | 513        |
| Customize the TCP/IP data sets or files . . . . .                   | 514        |
| Modify the change log inventory job. . . . .                        | 515        |
| Step 4: Populate the communications database . . . . .              | 516        |
| SYSIBM.LOCATIONS table . . . . .                                    | 516        |
| SYSIBM.IPNames table . . . . .                                      | 517        |
| SYSIBM.USERNames table . . . . .                                    | 518        |
| Step 5: Start TCP/IP support . . . . .                              | 518        |
| Step 6: Tuning TCP/IP. . . . .                                      | 519        |

---

## Chapter 13. 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.

The information in this section includes:

- “The database protocols (DRDA vs private)”
- “The communications protocols”
- “The role of the communications database (CDB)” on page 466
- “Enhancements since Version 5” on page 467
- “DB2 installation considerations” on page 467

See *DB2 Data Sharing: Planning and Administration* for information about connecting data base management systems (DBMSs) with a data sharing group.

---

### The database protocols (DRDA vs private)

Applications have two access methods to control remote access:

- The recommended method is to use **DRDA**.

With DRDA, the application connects to a server at another location and executes packages that have been previously bound at that server. The application uses a CONNECT statement, a 3-part name, or an alias (if bound with DBPROTOCOL (DRDA)) to access the server.

Queries can originate from any system or application that issues SQL statements as a *requester* in the formats required by DRDA.

Although use of Distributed Relational Database Architecture (DRDA) is not visible to you, information about it is available on the Open Group web site at [www.opengroup.org](http://www.opengroup.org). For two-phase commit using Systems Network Architecture (SNA) connections, DB2 supports both presumed abort and presumed nothing protocols that are defined by DRDA. 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*.

- The other method, *which is not recommended*, is to use **DB2 private protocol**.  
With private protocol, the application must use an alias or 3-part name to direct the SQL statement to a given location. Private protocol only works between requesters and servers that are both DB2 subsystems. Private protocol does not support many distributed functions, such as TCP/IP or stored procedures. The newer data types, such as LOB or user-defined types, are also not supported by private protocol.

---

### The communications protocols

DDF uses TCP/IP or SNA to communicate with other systems. Figure 81 on page 466 shows the connectivity options you have with DB2's DDF.

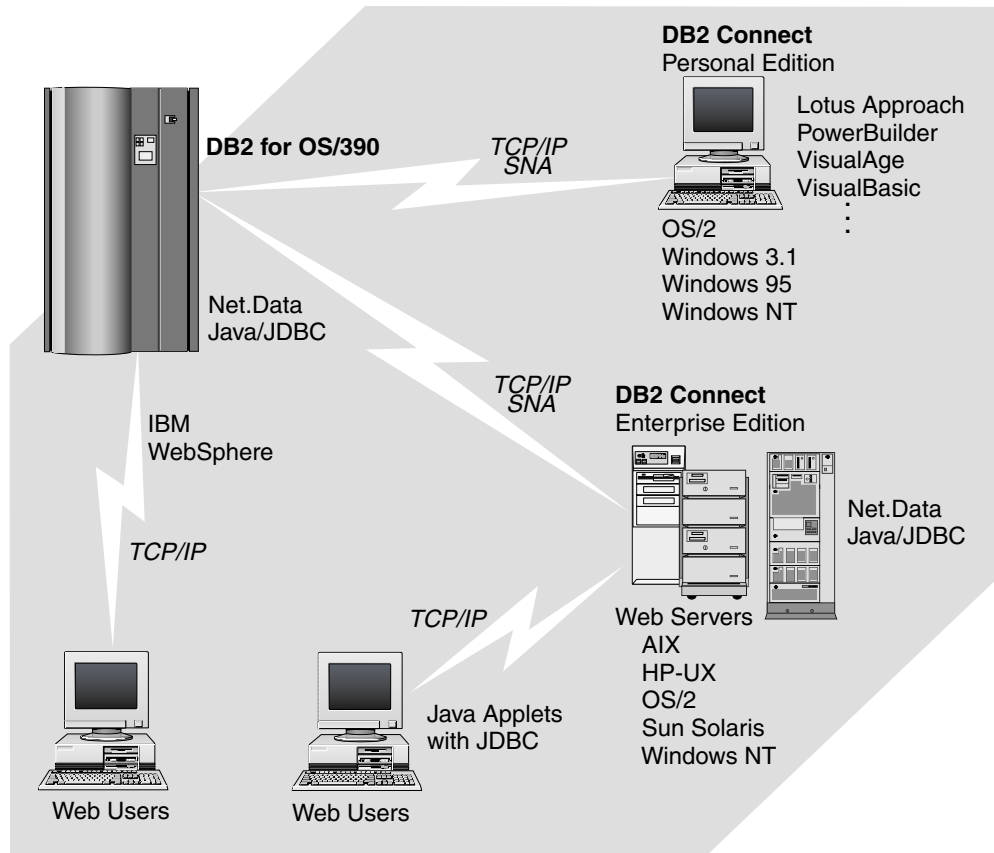


Figure 81. Connectivity options

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.

## The role of the communications database (CDB)

When sending a request, DB2 uses the LINKNAME column of the SYSIBM.LOCATIONS catalog table to determine which protocol to use, as shown in Figure 82 on page 467. To receive VTAM requests, you must select an LUNAME in installation panel DSNTIPR. To receive TCP/IP requests, you must select a DRDA port and a resynchronization port in installation panel DSNTIP5. TCP/IP uses the server's port number to pass network requests to the correct DB2 subsystem.

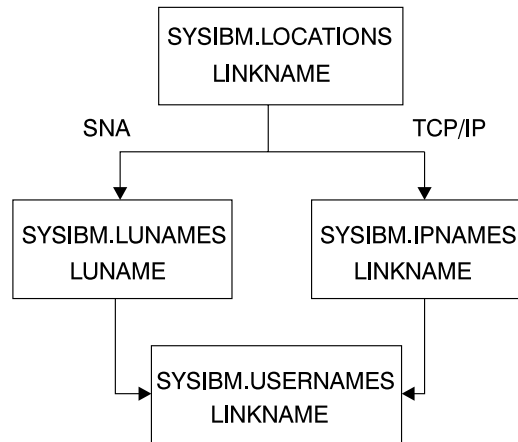


Figure 82. The linkname column of SYSIBM.LOCATIONS determines protocol

If the value in the LINKNAME column is found in the SYSIBM.IPNAMES table, TCP/IP is used for DRDA connections. If the value is found in SYSIBM.LUNAMES table, SNA is used. If the same name is in both SYSIBM.LUNAMES and SYSIBM.IPNAMES, TCP/IP is used to connect to the location.

**Attention:** A requester cannot connect to a given location using both SNA and TCP/IP protocols. For example, if your SYSIBM.LOCATIONS specifies a LINKNAME of LU1, and if LU1 is defined in both the SYSIBM.IPNAMES and SYSIBM.LUNAMES table, TCP/IP is the only protocol used to connect to LU1 from this requester for DRDA connections. For private protocol connections, the SNA protocols are used. If you are using private protocol connections, the SYSIBM.LUNAMES table must be defined for the remote location's LUNAME

---

## Enhancements since Version 5

Some of the DRDA enhancements since Version 5 include:

- Support for returning multiple extra DRDA query blocks on each network transmission. This support allows an application that fetches a large number of rows to minimize network overhead and utilize the full bandwidth of the network. (For details, see the EXTRA BLOCKS REQ and EXTRA BLOCKS SRV fields on “Distributed data facility panel: DSNTIP5” on page 210.
- A new bind option, DBPROTOCOL, which lets you use DRDA protocol even for applications that use 3-part names. This bind option lets your applications that currently use private protocol to begin taking advantage of DRDA protocol. If DBPROTOCOL is not specified, the default is the value you entered in the DATABASE PROTOCOL field on “Distributed data facility panel: DSNTIP5” on page 210.

**Recommendation:** Move all your distributed applications to DRDA. Private protocol is no longer being enhanced and will be discontinued in a future release. DRDA is strongly recommended for new applications. Consider the impact to existing applications before converting them to DRDA.

---

## DB2 installation considerations

The installation options for DDF are described in “Distributed data facility: DSNTIPR” on page 205 and “Distributed data facility panel: DSNTIP5” on page 210. Use these option to define, among other things:

- Whether you want DDF to start automatically when DB2 starts

- Important names for this DB2 subsystem, including a LU name and NETID  
Even if you do not plan to use VTAM, you still must define an LU name and NETID, because DB2 as a requester using TCP/IP protocols generates the unit of work using NETID and LUNAME.
- Thread management options
- Security options
- Default database protocol (DRDA or private)
- TCP/IP port numbers
- Control of the number of DRDA query blocks that can flow on a network request that was specified with OPTIMIZED FOR  $n$  ROWS where  $n$  exceeds the number of rows that fit in a single query block.

**Support for extended dynamic SQL:** If this DB2 subsystem services requesters that support extended dynamic SQL, such as SQL/DS, enter YES in field DESCRIBE FOR STATIC on installation panel DSNTIPF. This option lets applications from the requesting system execute SQL DESCRIBE statements that appear as extended dynamic SQL statements in the requesting system, but appear as static SQL in the DB2 package. For the option to take effect, you must bind the package with DESCRIBE FOR STATIC enabled.

**Test Your Connections** You should test systems with each other to ensure that their communications setups are correct. If you are testing with another DB2 for OS/390 and z/OS, enter the location name of that other site in field REMOTE LOCATION of installation panel DSNTIPY. The remote location must also have DDF installed and active and must have run the first sample job, DSNTJ1.

---

## Chapter 14. 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: Customize VTAM for DB2” on page 471**

To make monitoring of the network easier, consider installing NetView. For information about Netview, see *NetView Installation and Administration Guide*. For information about planning your network, see *Planning for NetView, NCP, and VTAM*. For information about installing VTAM, see *VTAM for MVS/ESA Network Implementation Guide*.

**“Step 2: Choose names and a password” on page 471**

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. Other systems use different terms for a location name. For example, DB2 Connect calls this the *target database name*. 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 the MVS system on which DB2 is running is part of an MVS Parallel 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 473**

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 499, in the data set DSN8VTAM in SDSNSAMP, and in examples throughout this chapter.

#### **“Step 4: Populate the communications database” on page 480**

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 483**

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 484**

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 492 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 495 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 or DRDA, at the other DB2 locations. For more information about creating aliases, see Part 2 (Volume 1) of *DB2 Administration Guide*.

#### **“Step 8: Provide Authorization for an Appropriate Level of Security”**

See Part 3 (Volume 1) of *DB2 Administration Guide* for information about security considerations for distributed data processing.



---

## Step 1: Customize VTAM for DB2

For DB2 to provide the best performance for distributed, you probably need to customize VTAM. Before you customize VTAM, consider the communication needs of your DB2 connections. Because you could allow your DB2 subsystem to send large amounts of data through the network, 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 507. 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*. (DB2 Connect refers to this as the *target database name*.) 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, tightly control the allocation of location names. 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 473.

- If 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. The generic name might be useful if your network is in a transitional period, and you want to use generic names to reference network nodes.

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.

See *DB2 Data Sharing: Planning and Administration* for instructions on setting up a generic LU name for a data sharing group.

## 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 Part 3 (Volume 1) of *DB2 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

**Location names and LU names:** When you populate the communications database (CDB) in the local DB2, you must know the location names (or DRDA RDBNAMs) and LU names of remote servers (that is, systems from which this DB2 will request data). The local DB2 does not need location names of requesters; however, you need to know the requesters' LU names if you intend to change default communication options.

DB2 does not receive DRDA RDBNAM from requesters other than DB2 for OS/390 and z/OS. If DB2 does not have an RDBNAM, it displays 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 480.

**Transaction program names (tpns):** If a server is not a DB2 for OS/390 and z/OS, it might have an additional name that uniquely identifies it. 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 for OS/390 and z/OS subsystem communicates with other DB2 for OS/390 and z/OS subsystems, you do not need to supply TPN values. The DB2 subsystems automatically choose the correct TPN values for both DRDA access and DB2 private protocol access.

**When a TPN is necessary:** You might need to supply TPN values when a DB2 subsystem requests data from a server that is not a DB2 for OS/390 and z/OS

subsystem. For cases where the server does not accept the default TPN for DRDA access, enter into your CDB the TPN chosen by that server. For DB2 for VM, for example, the TPN is the SQL database machine ID.

**TPN values accepted by DB2 for OS/390 and z/OS:** A requester that is not DB2 for OS/390 and z/OS 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 requesters enter the TPN as two separate fields: a 1-byte prefix (X'07') and a 3-byte suffix ('6DB').

## Names chosen by Spiffy Computer Company

Spiffy has chosen the location names and LU names shown in Table 117, some of which we use in later examples.

Table 117. 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 | DB2 for VM production system |
| USIBMSTOSQL2  | LUSQLDS | TPNSQLDS2 | DB2 for VM test system       |

**Note:** USIBMSTODB21 plans to accept requests from many OS/2 and Windows NT 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”
- A MODEENT macro, described in “The modeent macro” on page 477.

Samples of both the APPL and MODEENT macros are in the DSN8VTAM sample data set.

## 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 83 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 83. Example of a VTAM APPL definition statement

For your convenience, the APPL statement example is provided in data set DSN8VTAM, in the sample library, SDSNSAMP.

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:

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>name</i>    | The 1 to 8 character LU name you chose in “Step 2: Choose names and a password” on page 471. For their USIBMSTODB21 DB2 system, Spiffy uses LUDB21.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>AUTOSES</b> | <p>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.</p> <p>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.</p> <p>The default is 0. The suggested value is 1 or greater to ensure that VTAM informs DB2 if a session is inactivated.</p> <p>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.</p> |
| <b>DMINWNL</b> | <p>The minimum number of parallel sessions in which, if there is contention for a conversation, this local DB2 subsystem is the winner.</p> <p>The suggested value is one-half the value of DSESLIM, described below.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>DMINWNR</b> | <p>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.</p> <p>For more information about session negotiations, see “Interpreting CNOS messages” on page 496.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>DSESLIM</b> | <p>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.</p> <p>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</p>                                                                                                                                                                       |

they are requesting sessions that they never need. For information about how to calculate this number, see “Calculating session limits” on page 492.

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 489 for information about how to do that.

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>MODETAB</b> | 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 477.                                                                                                                    |
| <b>PRTCT</b>   | <p>If you decided to use a password, as described in “Step 2: Choose names and a password” on page 471, 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.</p> <p>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 Part 3 (Volume 1) of <i>DB2 Administration Guide</i>.</p>                                                       |
| <b>SECACPT</b> | <p>The level of conversation-level security allowed. <b>Recommendation:</b> Use 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 Part 3 (Volume 1) of <i>DB2 Administration Guide</i>.</p> <p>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.</p>                                                                                                      |
| <b>VERIFY</b>  | <p>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.</p> <p>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 more information about partner LU verification, see Part 3 (Volume 1) of <i>DB2 Administration Guide</i> and <i>VTAM for MVS/ESA Network Implementation Guide</i>.</p> <p>DB2 has no dependency on the value you choose.</p> |
| <b>VPACING</b> | 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 486.

### 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 Part 4 of *DB2 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.

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ACBNAME</b>  | The LU name for the DB2 subsystem. If the ACBNAME is different from the APPL name and both the originating and destination LUs are in the same VTAM domain, do not refer to the ACBNAME in a CDB definition. If the ACBNAME is not the same as the APPL name, VTAM may encounter name conflicts.                                                                                                                                                      |
| <b>DDRAINL</b>  | 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).                                                                                                                                                                                    |
| <b>DRESPL</b>   | 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).                                                                                                                                                                                                                                |
| <b>EAS</b>      | 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.                                                                                                                                                                                                                                                                                       |
| <b>LMIDENT</b>  | 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.                                                                                                                                                                                                                                                     |
| <b>MAXPVT</b>   | 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.                                                                                                                                                                                                        |
| <b>OPERCNOS</b> | <p>The ability to have a VTAM operator display and set VTAM session limits for a given LUNAME and MODENAME.</p> <ul style="list-style-type: none"><li>• 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.</li><li>• Use NALLOW, the default, to make sure VTAM operators are not able to change DB2's session limits dynamically.</li></ul> |

## 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 473

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 496) 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.
- IBMRDB is a recommended mode entry because it is used as a default for DRDA 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.
- IBMDB2LM is a recommended mode entry because it is used as a default for DB2 private protocol 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 84 on page 479 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. For your convenience, a sample MODEENT is included in data set DSN8VTAM, in SDSNSAMP. 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
        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'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
        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'06020000000000000000122F00' LU6.2 LU TYPE
MODEEND
END

```

Figure 84. 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.

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>name</i>    | The <i>name</i> option (IBMDB2LM and IBMRDB in the examples) is optional and has no function in the specification of a logon mode table.                                                                                                                                                                                                                                                                                                                   |
| <b>LOGMODE</b> | 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.                                                                                                                                                                                                                                                                                           |
| <b>TYPE</b>    | TYPE=0 indicates that DB2 is using a negotiable BIND, which is required for communicating with dependent LUs.                                                                                                                                                                                                                                                                                                                                              |
| <b>SRCVPAC</b> | Specifies the secondary receive pacing count. The DB2 suggested value is X'00'.                                                                                                                                                                                                                                                                                                                                                                            |
| <b>SSNDPAC</b> | 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.                                                                                                                                                                                                                                                                            |
| <b>RUSIZES</b> | 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 ( $8 \times 2^9$ ) across <i>at one time</i> , but there is no limit on how much total information can be sent. |

|                |                                                                                |
|----------------|--------------------------------------------------------------------------------|
| <b>FMPROF</b>  | This constant specifies the function management profile required for LU 6.2.   |
| <b>TSPROF</b>  | This constant specifies the transmission services profile required for LU 6.2. |
| <b>PRIPROT</b> | This constant specifies the primary LU protocols used in LU 6.2.               |
| <b>SECPROT</b> | This constant specifies the secondary LU protocols used in LU 6.2.             |
| <b>COMPROT</b> | This constant specifies the common LU protocols used in LU 6.2.                |
| <b>PSERVIC</b> | 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 486.

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 and Windows NT 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 *DB2 Data Sharing: Planning and Administration* for more information. Part 3 (Volume 1) of *DB2 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 *DB2 Application Programming and SQL Guide*. For instructions on granting privileges to users on remote DB2 subsystems, see Part 3 (Volume 1) of *DB2 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:

|                          |                                                                                                                                                                                         |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>LOCATION CHAR(16)</b> | 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.If the remote LU exists in the same VTAM domain, specify the APPL name, not the ACBNAME.

**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 471.

Spiffy’s USIBMSTODB21 location wants a LOCATIONS table that looks like Table 118.

Table 118. 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. If the remote LU exists in the same VTAM domain, specify the APPL name, not the ACBNAME.

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 Kerberos 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 Part 3 (Volume 1) of *DB2 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 488.

#### **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 Part 3 (Volume 1) of *DB2 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 a blank indicates that the name specified in the LUNAME column is to be used. See Chapter 4 of *DB2 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 119.

Table 119. Spiffy's SYSIBM.LUNAMES table. The row of blanks is a default row that Spiffy intends to use for Windows NT and OS/2 requesters in its initial testing.

| LUNAME   | SYSMODENAME | USERSECURITY <sup>1</sup> | ENCRYPTPSWDS | MODESELECT | USERNAMES |
|----------|-------------|---------------------------|--------------|------------|-----------|
| LUDB21   |             |                           |              |            |           |
| LUDB22   |             |                           |              |            |           |
| LUSQLDS  |             |                           |              |            |           |
| (blanks) |             |                           |              |            |           |

Note: <sup>1</sup> USERSECURITY 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

SYSIBM.USERNAMES contains information needed for outbound and inbound ID translation and also for *come from* checking. See Part 3 (Volume 1) of *DB2 Administration Guide* for information about populating this table.

---

## 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 473 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 477.

---

## 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 Part 5 (Volume 2) of *DB2 Administration Guide* for more information about monitoring and tuning DB2 distributed applications.

In this section we describe:

- “Controlling buffer storage” on page 485.  
By sending large amounts of data through the network, DB2 can cause problems with your VTAM I/O buffer pool.
- “Controlling pacing” on page 486.  
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 487  
Consider modifying session limits if you have problems obtaining enough sessions to handle your distributed workload efficiently.
- “Modifying class of service” on page 488.  
Specifying a class of service can help you assign priorities to your network applications.
- “Associating applications with modes” on page 488.  
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 120 summarizes the relationship.

Table 120. Relationship between DB2's CDB and VTAM macros

| Macro Name | Option      | CDB table.column                                                                                                     | Relationship                                                                                                 |
|------------|-------------|----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| APPL       | <i>name</i> | LOCATIONS.LINKNAME<br>LUNAMES.LUNAME<br>LUMODES.LUNAME<br>MODESELECT.LUNAME<br>USERNAMES.LINKNAME<br>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 495.

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. Recommendation: 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” on page 486 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 487 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 477.

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 473. You set the RU size in the MODEENT macro described in “The modeent macro” on page 477. 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 85. 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.

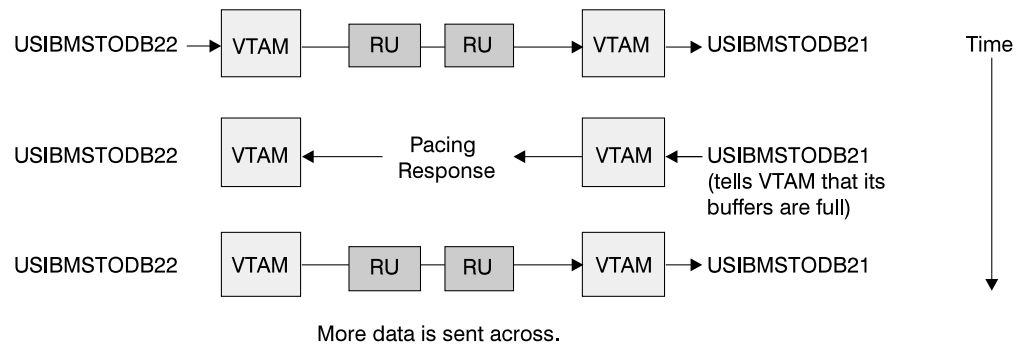


Figure 85. How pacing works

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.

**Recommendation:** Start 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  $2 \times 4\text{KB} = 8\text{KB}$  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 495 for information about calculating buffer pool sizes.

### Recommendation for MODEENT pacing options

The MODEENT macro described on 477, contains several pacing options:

- |                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PSNDPAC</b> | This does not apply to DB2; therefore, you can ignore this option.                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>SSNDPAC</b> | 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. <b>Recommendation:</b> Specify a nonzero value for this option.                                                                                                                                                                                                                                     |
| <b>SRCVPAC</b> | <p>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.</p> <p>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.</p> |

## 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 492.

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 492.

## **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 491 is General-use Programming Interface, as defined in “Notices” on page 541.

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 491.

### **Update LUNAMES to associate modes with LU names**

This table is described in “Step 4: Populate the communications database” on page 480. 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); not the recommended connection type
- SQL processing conversations (can be many for DB2 private protocol access; only one for DRDA access)

Figure 86 on page 489 shows how these different conversations are used for a DB2 private protocol access application.

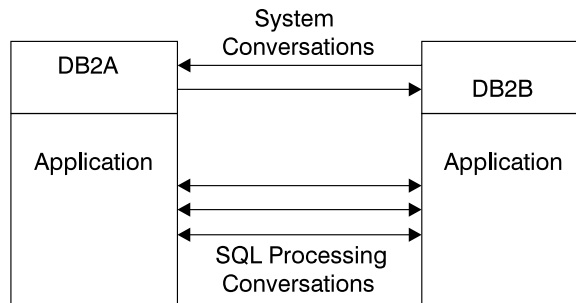


Figure 86. Conversations used for DB2 private protocol access

**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 121.

Table 121. Spiffy's LUNAMES table, after update

| LUNAME | SYSMODENAME | USERSECURITY <sup>1</sup> | ... | MODESELECT | ... |
|--------|-------------|---------------------------|-----|------------|-----|
| LUDB22 | LOC2MODE    |                           |     | Y          |     |

**Note:** <sup>1</sup>USERSECURITY 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 496.

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 is overrides the number in the DSESLIM option of the VTAM APPL definition statement during CNOS processing, as described in “Interpreting CNOS messages” on page 496.

#### **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 122 shows the search order of the MODESELECT table.

*Table 122. Precedence search order for MODESELECT table of CDB*

| <b>AUTHID</b> | <b>PLANNAME</b> | <b>Result</b>                                                                           |
|---------------|-----------------|-----------------------------------------------------------------------------------------|
| 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 USER NAMES 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 493.

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, ... A<sub>N</sub>.
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, ... C<sub>n</sub>, 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  $C1+C2+ \dots C_N=LIMIT\_A$ .
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**

Has 3 cursors open  
concurrently referencing  
location B.

**App\_2**

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.

**App\_3**

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 492 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 87 on page 494.

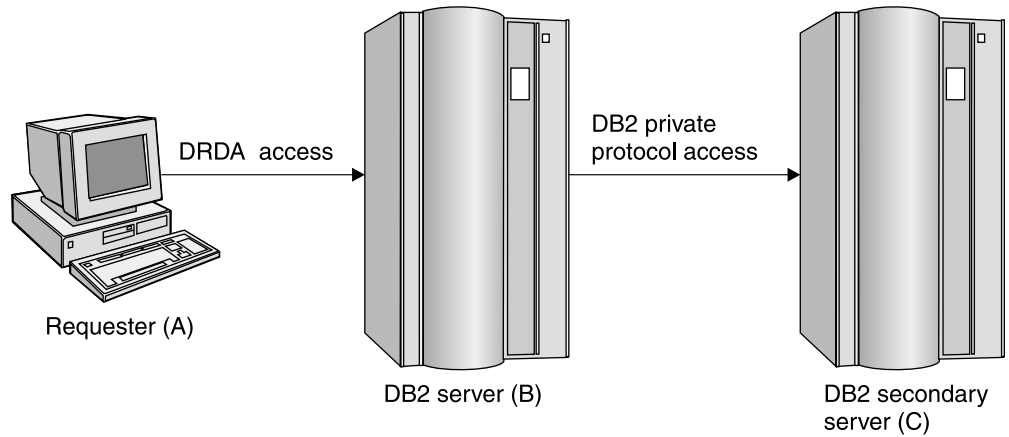


Figure 87. Example of a mixed application

The session requirements for such an application at each location are:

- (A) For the requester, the requirement is 1, which is the only number of sessions that can be used by an DRDA access application.
- (B) The requirement is the value for DB2 private protocol access (connection from B to C) plus 1 for the DRDA access connection from B to A.
- (C) The requirement is the value for DB2 private protocol access only. No DRDA access is possible at this site for this application.

- An application uses DRDA access to connect to one location, then drops that connection using SQL RELEASE and COMMIT statements, then uses DB2 private protocol access to access another DB2. (This same method could be used to access the same location; you would still have to drop your DRDA access connection to access DB2 using DB2 private protocol access.)

This type of application is shown in Figure 88 on page 495.



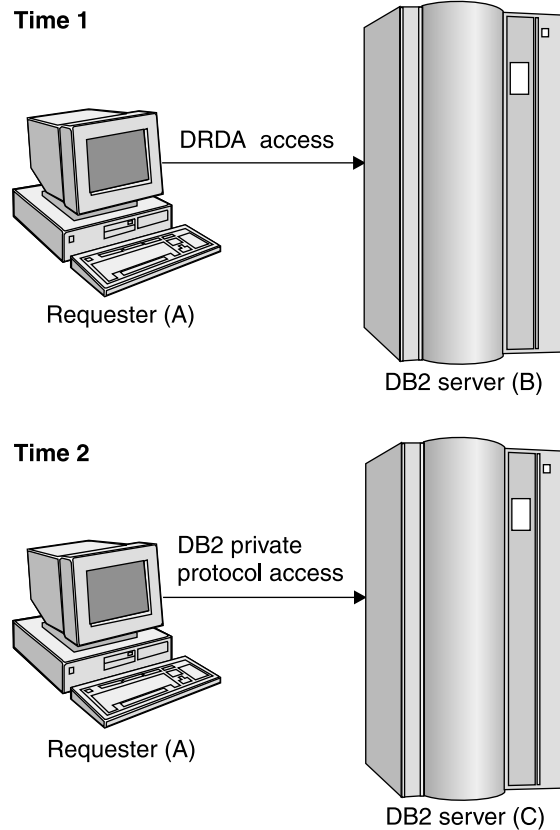


Figure 88. Another example of a mixed application

The session requirements for such an application at each location are:

- (A) The session requirement is the same as for DB2 private protocol access because only one type of access can be active at a time.
- (B) The requirement is 1, for the DRDA access connection.
- (C) The requirement is the value for DB2 private protocol access only.

## 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.

$$\text{PIUBUF} = \text{CEILING}((29 + \text{RUSIZE}) / \text{BUFSIZE})$$

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 502 and “NCP-connected DB2s” on page 503). 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 486, 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 492 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:

| USIBMSTODB21<br>APPL Values |    |
|-----------------------------|----|
| DSESLIM                     | 50 |
| DMINWNL                     | 25 |
| DMINWNR                     | 25 |
| AUTOSSES                    | 1  |

| USIBMSTODB22<br>APPL Values |    |
|-----------------------------|----|
| DSESLIM                     | 40 |
| DMINWNL                     | 20 |
| DMINWNR                     | 20 |
| AUTOSSES                    | 1  |

| USIBMSTODB21 SYSIBM.LUMODES |          |           |
|-----------------------------|----------|-----------|
| LUNAME                      | MODENAME | CONVLIMIT |
| LUDB22                      | SYSTOSYS | 2         |
| LUDB22                      | IBMDB2LM | 80        |

| USIBMSTODB22 SYSIBM.LUMODES |          |           |
|-----------------------------|----------|-----------|
| LUNAME                      | MODENAME | CONVLIMIT |
| LUDB21                      | SYSTOSYS | 2         |
| LUDB21                      | IBMDB2LM | 50        |

Figure 89. CNOS negotiation example: VTAM and DB2 definitions

Assume that USIBMSTODB21’s DDF is started first. CNOS processing fails because USIBMSTODB22’s DDF has not yet started, and you get a message at the console. When USIBMSTODB22’s DDF is started, CNOS processing can begin.

USIBMSTODB21 sends to USIBMSTODB22 a CNOS value of 80, which is its CONVLIMIT value. However, USIBMSTODB22 replies with a value of 40 (its DSESLIM value), and, as shown in Figure 90 on page 498, that becomes the negotiated value for the CNOS started by USIBMSTODB21. Both systems begin starting the number of sessions specified in their respective AUTOSSES options.

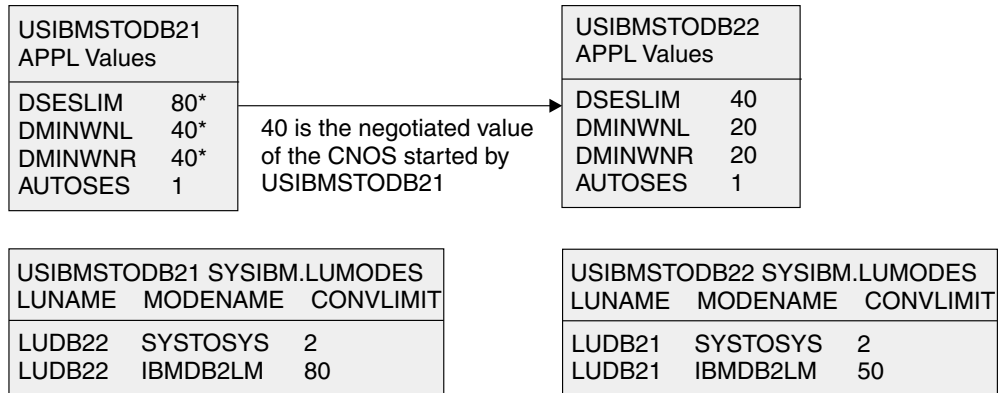


Figure 90. Result of CNOS negotiation started by USIBMSTODB21. Overridden values are noted with asterisks (\*).

VTAM does not start all 40 sessions unless the two AUTOSES values total up to 40 or greater. Instead, VTAM delays starting the other sessions until they are needed.

Everything up to this point occurred because USIBMSTODB21 issued CNOS. Now, as shown in Figure 91, USIBMSTODB22 starts CNOS processing back to USIBMSTODB21 because its CDB has CNOS limits specified (50 in CONVLIMIT). USIBMSTODB21's VTAM sees that DB2 allows up to 80, so VTAM sends the CNOS reply message back to USIBMSTODB22 unchanged (50). USIBMSTODB22's CONVLIMIT value of 50 is compared with USIBMSTODB21's overridden value of 80 from the previous CNOS, and 50 is chosen as the value.

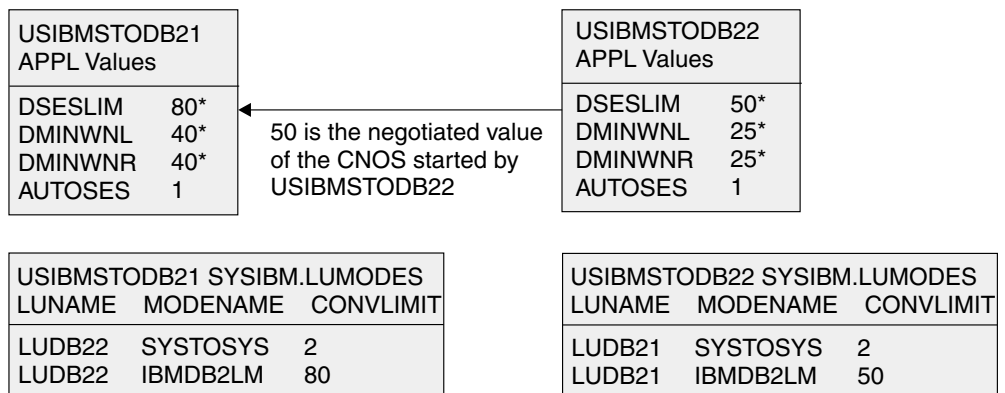


Figure 91. CNOS negotiation from USIBMSTODB22 to USIBMSTODB21. Overridden values are noted with asterisks (\*).

If the new negotiated value is smaller than the number already started by USIBMSTODB21, then VTAM terminates the number of sessions that makes up the difference. If the CONVLIMIT value at USIBMSTODB22 is 20, for example, VTAM terminates 20 sessions on behalf of the request from USIBMSTODB22 because the lowest negotiated value always wins. If a session is currently being used by a conversation, the session is terminated as soon as the conversation is deallocated.

---

## Sample VTAM definitions to connect two DB2s

These definitions are included to give you some guidance on setting up your network to connect two DB2s. It is not intended to give you information about all the options; the ones most relevant have been discussed previously in this chapter. We suggest you see VTAM publications for more information about VTAM options.

This set of definitions includes the basic definitions you need to connect two DB2s. Additional options for channel-to-channel and Network Control Program (NCP) connections are covered as well.

### Basic definitions

The basic definitions here are required for all VTAM connections. For more information about these definitions, see *VTAM for MVS/ESA Resource Definition Reference*.

```

*****
**      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
          PRTCT=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
          PRTCT=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
                FMPROF=X'13',                 X
                TSPROF=X'07',                 X
                PRIPROT=X'B0',                 X
                SECPROT=X'B0',                 X
                PSERVIC=X'060200000000000000000300', X
                COMPROT=X'D0B1',               X
                RUSIZES=X'8585',               X
                ENCR=B'0000'
*****

```

Figure 92. Basic VTAM definitions (Part 1 of 2)

```

** DB2 DEFAULT MODE FOR SYSTEM 1 AND SYSTEM 2 PRIVATE PROTOCOL          **
*****
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
                    TS PROF=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 DRDA 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
                    TS PROF=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
                    TS PROF=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 92. Basic VTAM definitions (Part 2 of 2)

## 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    *
*****
MVSD2   PATH DESTSA=2,ER1=(2,1),VR1=1,              X
                                   VRPWS10=(2,30),VRPWS11=(2,30),VRPWS12=(2,30)
*****
**          PATH - NETWORK ROUTES FOR SYSTEM 2    *
*****
MVSD1   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 93. Channel-to-channel (CTC) definitions (Part 1 of 2)



```

*****
**      CDRM DEFINITIONS FOR SYSTEM 1 AND 2 (SAME DEFINITION USED)      *
*****
          VBUILD TYPE=CDRM
DB1CDRM CDRM SUBAREA=1, ISTATUS=ACTIVE, CDRSC=OPT
DB2CDRM CDRM SUBAREA=2, ISTATUS=ACTIVE, CDRSC=OPT

*****
**      ATCCONTA - NETWORK CONFIGURATION LIST FOR SYSTEM 1              *
*****
DB1PATH,DB1CTC,DB1RSC,DB1APPLS,DBCDRMS
*****
**      ATCCONTB - NETWORK CONFIGURATION LIST FOR SYSTEM 2              *
*****
DB2PATH,DB2CTC,DB2RSC,DB2APPLS,DBCDRMS

```

Figure 93. Channel-to-channel (CTC) definitions (Part 2 of 2)

## NCP-connected DB2s

The Advanced Communications Facility/Network Control Program (ACF/NCP) is a product you use to generate a network control program load module, which is loaded from the host into a communications controller. The network control program controls the lines and devices attached to it. It transfers data to and from the devices and handles any errors that occur, including retries after line errors.

A communications controller can be locally attached to a host via a channel, or it can be link-attached to another communications controller that is channel-attached.

Our sample definitions are used for the following setup:



When you are defining your NCP connections, remember the following:

- MAXBFRU must be large enough to handle the biggest PIU that is sent to the NCP. In our example, DB2 is sending 4125 bytes per PIU (4096 + a 29-byte network header). Given an IOBUF buffer size of 441 bytes, MAXBFRU must therefore be at least 10 ( $10 \times 441 = 4410$ , which is greater than 4125).
- The MAXDATA option must also be large enough to handle biggest PIU (RUSIZE + 29 bytes).

If DB2 is using existing NCP definitions, you should make sure your MAXBFRU and MAXDATA options are large enough. If these values are too small, you get an SNA X'800A' sense code, indicating that the PIU was truncated during transmission.

```

*****
*           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 94. Network control program (NCP) definitions (Part 1 of 4)

```

*****
*      BUILD MACRO SPECIFICATIONS      *
*****
ACFNCPBD BUILD BFRS=(240),          NCP BUFFER SIZE,# EP BUFFERS      X
      BRANCH=1000,                  X
      CATRACE=(YES,10),              X
      CSMHDR=27F5C711C3F0405C40C8C4D9405C,          X
      CSMHDC=40E3C5E7E3405C5C,          X
      CSMSC=5C5C40E5E3C1D440E2C8E4E3C4D6E6D540,      X
      CMSGC=6040C8C1E240C2C5C7E4D5405C5C,          X
      DIALTO=60,          WAIT 1 MIN FOR AUTOCALL ANSWER      X
      DR3270=NO,          NO DYNAMIC RECONFIG              X
      DSABLTO=3.0,        TIME TO DETECT DSR DROP          X
      ENABLTO=2.2,        TIME TO DETECT DSR AFTER ENABLE    X
      LOADLIB=NCPLD,      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 94. Network control program (NCP) definitions (Part 2 of 4)

```

*****
**
*          SYSCNTRL OPTIONS - REQUIRED BY VTAM          *
**
*****
          SYSCNTRL OPTIONS=(ENDCALL,MODE,RCNTRL,RCOND,RECMD,RIMM,
          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,
              UNITSZ=441           = BUFSIZE IN IOBUF FOR VTAM      X

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,
              UNITSZ=441           = BUFSIZE IN IOBUF FOR VTAM      X

*****
*          PATH STATEMENTS          *
*****
          PATH DESTSA=3,          SYS1          X
              ER4=(3,1),          SYS1

          PATH DESTSA=4,          SYS2          X
              ER4=(4,1),          SYS2

```

*Figure 94. Network control program (NCP) definitions (Part 3 of 4)*

```

*****
*
*   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 94. Network control program (NCP) definitions (Part 4 of 4)

## 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 Part 3 of *DB2 Utility Guide and Reference*.

---

## Chapter 15. 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 UNIX System Services.

**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 requester. 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: Prepare the Language Environment runtime library” on page 512.
- “Step 2: Enable DDF for UNIX System Services” on page 513.
- “Step 3: Define the DB2 subsystem to TCP/IP” on page 513.
- “Step 4: Populate the communications database” on page 516.
- “Step 5: Start TCP/IP support” on page 518.
- “Step 6: Tuning TCP/IP” on page 519.

**Note:** DB2 obtains more information from the TCP/IP stack, therefore requiring more configuration than many other daemons

If you do not use VTAM to communicate with remote sites, you still must define VTAM to DB2 as described in “Chapter 14. Connecting systems with VTAM” on page 469 because DB2 TCP/IP communications uses NETID and LUNAME to identify units of work.

See Chapter 4 of *DB2 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. 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). See Figure 81 on page 466 for connection examples.

## TCP/IP limitations

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 subsystem parameter defines the minimum security requirements for all TCP/IP clients because inbound security requirements cannot be established on individual clients. See the description of the TCP/IP ALREADY VERIFIED field on installation panel DSNTIP5 (“Distributed data facility panel: DSNTIP5” on page 210).
- You cannot use the CDB for *come from* checking of TCP/IP clients.

## Initializing a TCP stack for use with a VIPA

When initializing a TCP stack for use with a virtual IP address (VIPA), the PROFILE.TCPIP configuration should contain the HOME list with the VIPA followed by the PRIMARYINTERFACE statement pointing at the appropriate VIPA to be used by DB2. It is not recommended to change the stack’s HOME list or the PRIMARYINTERFACE while DDF is started. The PRIMARYINTERFACE and the stack’s HOME list may be changed (with the VARY OBEY command) while the stack is running without recycling the stack. If a new HOME list is specified with the VARY OBEY file, the PRIMARYINTERFACE should also be specified.

## 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, domain names, and resync IP addresses that are part of the client’s unit of work. DB2 Connect V5 utilizes this support by allowing DB2 to act as its Transaction Manager Database (TM\_DATABASE), thereby eliminating the need to have a local database to manage the 2-phase commit process. 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 and z/OS, 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.

## Multiple DB2 subsystems on a single OS/390 image

OS/390 Version 2 Release 8 has introduced configuration improvements such that it is no longer a recommended configuration to run multiple TCP/IP stacks with DB2.

TCP scalability and reliability are such that running two stacks may cause problems due to increased storage and CPU consumption than running one stack, with few or no expected benefits. These configuration improvements are especially helpful when there are multiple DB2 subsystems installed within a data sharing group on the single OS/390 image, but these improvements can be useful in a non-data sharing environment.

Although multiple TCP/IP stacks are not recommended for DB2, there may be reasons why multiple stacks may be appropriate for your installation. A possible reason for using multiple TCP/IP stacks could be to separate internet and intranet traffic. The firewall on the stack handling external traffic can be configured only to forward very selected traffic internally, whereas the internal stack does whatever is indicated with internal traffic, including forwarding to the "external" stack. Second, a denial-of-service attack on the stack handling external traffic will not affect traffic on the stack handling internal traffic, so there is an element of resistance to attack with two stacks.

### **Multiple DB2 subsystems with multiple TCP/IP stacks**

OS/390 UNIX System Services provides a way to configure DB2 to use a single transport stack. This allows each DB2 subsystem to be restricted to a single stack. Each would be reachable via any IP address owned by the selected stack, but not via the other stack(s). Adding BPXTCAFF as a new job step for the *ssnmDIST* procedure allows DB2 to bind to a single stack when there are at least two stacks running on a single OS/390 image.

### **Multiple DB2 subsystems with one TCP/IP stack**

The TCP Server Bind Control allows multiple DB2 subsystems to use the same port number within the same OS/390 image. This allows multiple DB2 subsystems that bind to any IP address (also known as INADDR\_ANY in sockets API terms) and the same port number on the same stack to be bound to separate IP addresses. For example, consider two DB2 subsystems in a data sharing group where both have the same LOCATION and same DRDA PORT. A new 'BIND ipaddr' parameter is added to the PORT statement. When DB2 (identified by job name *ssnmDIST*) issues a bind to the port number in the PORT statement and to INADDR\_ANY, the bind is restricted to the IP address specified on the PORT statement for that DB2 subsystem. This allows two DB2s to share the same stack by limiting them to different single IP addresses (virtual IP addresses are recommended). This allows them to be reached by any physical connectivity that can get to the stack.

---

## **Step 1: Prepare the Language Environment runtime library**

Because DDF uses some functions in the Language Environment library, DDF needs access to the runtime library. The standard way to handle this is to include the Language Environment library in a STEPLIB concatenation for the DDF JCL procedure. The Language Environment 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 Language Environment library in the MVS link list, which does not require the library to be APF authorized. If you choose this alternative method, remove the Language Environment library concatenation from the DDF JCL procedure.

---

## Step 2: Enable DDF for UNIX System Services

DDF uses the UNIX System Services 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 UNIX System Services environment as a superuser to perform asynchronous I/O socket calls. If the socket call fails because of insufficient authorization (reason code=1148033C) refer to message DSNL512 in *DB2 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 UID 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 UNIX® System Services group ID setting. You define RACF groups to be UNIX System Services groups with one of the following commands:

```
ADDGROUP ddfgid OMVS(GID(x))...  
ALTGROUP ddfgid OMVS(GID(x))...
```

where *x* is any valid, unique identifier.

Replace *ddfgid* in the above commands with the MVS user 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 *OS/390 Security Server (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. *DB2 Data Sharing: Planning and Administration* has information for assigning port numbers for systems that have enabled data sharing.

Figure 95 on page 514 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.

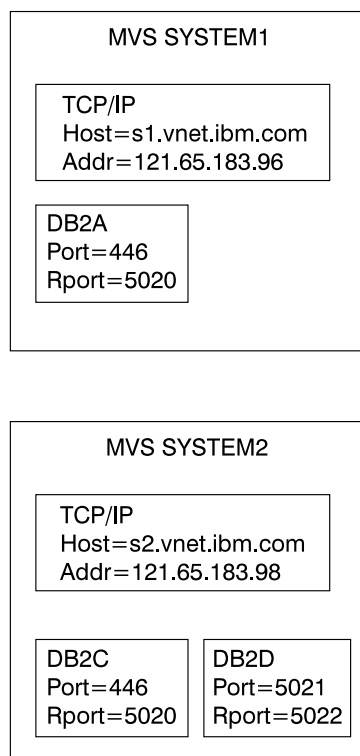


Figure 95. Typical MVS configurations

## Customize the TCP/IP data sets or files

Follow these steps to customize your TCP/IP data sets or files. **UNIX System Services should already be installed.** See the DB2 Program Directory for required maintenance levels. For details on customizing your TCP/IP data sets see *OS/390 UNIX System Services Planning*.

1. Find the TCPIP.TCPIP.DATA data set.

This data set defines the high level qualifier (*hlq*) which is added to the beginning of other data set names used by TCP/IP.

2. Find the *hlq*.TCPPARMS(PROFILE) data set.

This data set contains the PORT statement used to make DRDA and resync port reservations. The following example from Figure 95 shows a sample *hlq*.TCPPARMS(PROFILE) entry for SYSTEM2:

```

PORT 446 TCP DB2CDIST          ; DRDA SQL port for DB2C
PORT 5020 TCP DB2CDIST          ; Resync port for DB2C

PORT 5021 TCP DB2DDIST          ; DRDA SQL port for DB2D
PORT 5022 TCP DB2DDIST          ; Resync port for DB2D

```

This example assumes that *DB2CDIST* and *DB2DDIST* are the DB2 started procedure names.

### 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.

**Recommendation:** To support a Parallel Sysplex environment or multiple network interfaces on a single host, use the DNS or *hlq*.HOSTS.LOCAL data set to associate multiple IP addresses with a single host name.

The following example from Figure 95 on page 514 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 518 is General-use Programming Interface, as defined in “Notices” on page 541.

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 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 14. Connecting systems with VTAM” on page 469). If RACF PassTickets are used, the LINKNAME must match the LUNAME of the remote site. Part 3 (Volume 1) of *DB2 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 *DB2 Application Programming and SQL Guide*. For instructions on granting privileges to users on remote DB2 subsystems, see Part 3 (Volume 1) of *DB2 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 123. The location USIBMSTODB21 uses the default DRDA PORT, 446.

Table 123. 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, UNIX System Services and TCP/IP must be started and the local host name must be defined in */etc/hosts*, *TCPIP.HOSTS.LOCAL*, or the domain name server (DNS).

DDF executes the following steps when it is started:

1. Issues a `gethostid` to obtain the TCP/IP stack's primary interface IP address.

The IP address can be determined by issuing a `"netstat home"` command. In the output, the primary interface is marked with a P in the Flg column. For example:

```
EZZ2350I MVS TCP/IP NETSTAT CS V2R8      TCPIP NAME: TCPIP      19:02:54
EZZ2700I Home address list:
EZZ2701I Address      Link      Flg
EZZ2702I -----
EZZ2703I 10.2.1.21     TR1
EZZ2703I 10.2.2.21     TR2
EZZ2703I 10.1.1.22     CTC02L
EZZ2703I 10.1.1.18     MPC00L
EZZ2703I 10.1.1.2      ENDER2L
EZZ2703I 10.2.10.21    FENET1
EZZ2703I 192.2.30.21   ATMLEC1
EZZ2703I 10.1.1.253    C1T0C2
EZZ2703I 10.1.100.21   LVIPA01      P
EZZ2703I 127.0.0.1     LOOPBACK
```

2. Issues a `gethostname` to register the HOST to WLM. TCP/IP returns the HOSTNAME found in the stack's `TCPIP.DATA` file
3. Issues a `gethostbyaddr` using the local IP address obtained from the `gethostid` to obtain the fully qualified domain name. In a data sharing environment, this domain name is used to generate the location and member specific domain names. The domain name is returned to clients to be used to open a connection to DB2 to perform resynchronization after a failure.
4. Issues `BIND` using `INADDR_ANY` to accept connections from any IP address supported by the host.



5. Issues `set_sock_opt` for all inbound and outbound connections for the following options:
- `SO_KEEPALIVE` to enable keepalive processing.
  - `SO_REUSEADDR` to allow the use of the same port.
  - `SO_SNDBUF` to set the send buffer size if less than 64K.

To support data sharing workload balancing and commit resynchronization, DDF requires the fully qualified host name to be established during startup processing. During startup, DDF issues a `gethostid` and `gethostbyaddr` socket request. The `gethostid` service obtains the local IP address and then a `gethostbyaddr` service is issued to get the fully qualified host name. The `gethostbyaddr` service attempts to resolve the hostname through a nameserver. If the nameserver is not present, or does not have an answer, then the local host tables are used. The local host tables search order for the resolver as follows:

1. `/etc/resolv.conf` file
2. `jobname.TCPIP.DATA`
3. `SYS1.TCPPARMS(TCPDATA)`
4. `TCP $hlq$ .TCPIP.DATA`

---

## 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, five 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 attempt to connect to each IP address has timed out.



---

## Part 4. Appendixes



---

## Appendix A. 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 524
- “Unicode support” on page 527
- “When remote packages should be rebound” on page 531

---

### 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, ASCII, and UNICODE are ways of encoding data. Character data transmitted from one DBMS to another might need to be converted to a different coded character set.

The UNICODE character encoding standard is a character encoding scheme that includes characters from almost all the living languages of the world. DB2 supports two implementations of the UNICODE encoding scheme: UTF-8 (a mixed-byte form) and UTF-16 (a double-byte form). For more information on UNICODE, see “Unicode support” on page 527.

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 the following facts:

- When you install DB2 you must specify a CCSID for DB2’s character data if:
  - you specify AUTO or COMMAND for the DDF STARTUP OPTION field on panel DSNTIPR.
  - your system will have any ASCII or UNICODE data, or EBCDIC mixed character or graphic data. In this case the MIXED DATA field of panel DSNTIPF must be YES, and the CCSID that you specify is the mixed CCSID for the encoding scheme.

Which CCSID you specify depends on the national language you use.

“Specifying a system coded character set identifier” on page 524 lists the choices available.

- There are three methods for character conversion support. The methods are searched in the following order:
  - DB2 catalog table SYSIBM.SYSSTRINGS.
    - If DB2 does not provide a conversion table for a certain combination of source and target CCSIDs you will get an error message, unless a conversion can be found by another method..
    - 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 *DB2 SQL Reference* explains those rules.
  - OS/390 Support for Unicode. You need OS/390 Version 2 Release 8 plus OS/390 Version 2 Release 8 Support for Unicode, or a later release of OS/390 (i.e. R9+), to use the conversion services.
  - Language Environment. You need OS/390 Version 2 Release 9 to use Language Environment.

---

## 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 Reference and Registry*. For general information about character conversion, see *Character Data Representation Architecture Overview*.

The CCSID of character strings at your site is determined by the CCSID you specify on 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 panel DSNTIPF, you can use any valid SBCS CCSID in the EBCDIC CODED CHAR SET and ASCII CODED CHAR SET fields. Table 124 lists a selection of common SBCS CCSIDs that might be used as source or target CCSIDs for EBCDIC or ASCII data. DB2 does not support the storing of data into all of these CCSIDs. That is, not all of the numbers listed in Table 124 are supported as target CCSIDs in conversion. To find out which combinations DB2 supports with SYSSTRINGS, issue the following SQL statement:

---

### General-use Programming Interface

```
SELECT * FROM SYSIBM.SYSSTRINGS;
```

---

### End of General-use Programming Interface

---

If OS/390 V2R9 is installed, refer to OS/390 C/C++ Programming Guide for additional conversions that are supported.

For more information about UNICODE and UNICODE CCSIDs, see “Unicode support” on page 527.

Table 124. Single-byte coded character set identifiers (CCSIDs)

| 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*    |
| Bosnia and Herzegovina (Latin)    | 870       | 852      | 912       | 1250/5346*    |
| Brazil (U.S. English)             | 37/1140*  | 850/858* | 819       | 1252/5348*    |
| Bulgaria (Cyrillic Multilingual)  | 1025      |          |           | 1251/5347*    |

Table 124. Single-byte coded character set identifiers (CCSIDs) (continued)

| Country or National Language | EBCDIC     | ASCII PC | ASCII AIX | ASCII WINDOWS |
|------------------------------|------------|----------|-----------|---------------|
| Canada (U.S. English)        | 37/1140*   | 850/858* | 819       | 1252/5348*    |
| Croatia                      | 870        | 852      | 912       | 1250/5346*    |
| Czech Republic               | 870        | 852      | 912       | 1250/5346*    |
| Denmark                      | 277/1142*  | 850/858* | 819       | 1252/5348*    |
| Finland (Swedish)            | 278/1143*  | 850/858* | 819       | 1252/5348*    |
| France                       | 297/1147*  | 850/858* | 819       | 1252/5348*    |
| Germany                      | 273/1141*  | 850/858* | 819       | 1252/5348*    |
| Greece                       | 875 or 423 | 869      | 813       | 1253/5349*    |
| Hungary                      | 870        | 852      | 912       | 1250/5346*    |
| Iceland                      | 871/1149*  | 850/858* | 819       | 1252/5348*    |
| International Latin-1        | 500/1148*  |          |           |               |
| Israel                       | 424        | 862      | 916       | 1255/5351*    |
| Italy                        | 280/1144*  | 850/858* | 819       | 1252/5348*    |
| Latin America (Spanish)      | 284/1145*  | 850/858* | 819       | 1252/5348*    |
| FYR Macedonia                | 1025       |          |           | 1251/5347*    |
| Netherlands (U.S. English)   | 37/1140*   | 850/858* | 819       | 1252/5348*    |
| New Zealand (U.S. English)   | 37/1140*   | 437      | 819       | 1252/5348*    |
| Norway                       | 277/1142*  | 850/858* | 819       | 1252/5348*    |
| Poland                       | 870        | 852      | 912       | 1250/5346*    |
| Portugal (U.S. English)      | 37/1140*   | 850/858* | 819       | 1252/5348*    |
| Serbia/Montenegro            | 1025       |          |           | 1251/5347*    |
| Spain                        | 284/1145*  | 850/858* | 819       | 1252/5348*    |
| Sweden                       | 278/1143*  | 850/858* | 819       | 1252/5348*    |
| Switzerland                  | 500 /1148* | 850/858* | 819       | 1252/5348*    |
| Thailand                     | 838        |          |           |               |
| Turkey (Latin 5)             | 1026       | 857      | 920       | 1254/5350*    |
| United Kingdom               | 285/1146*  | 850/858* | 819       | 1252/5348*    |
| 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=NO on installation panel DSNTIPF, then specify an SBCS CCSID from Table 125 on page 526 in the EBCDIC CCSID field on DSNTIPF. If your system uses ASCII data, then specify a SBCS CCSID from Table 126 on page 526 in the ASCII CCSID field on DSNTIPF. Mixed character data and graphic data cannot be defined on such a system.

If you specified MIXED DATA=YES on installation panel DSNTIPF, then specify a mixed CCSID from Table 125 in the EBCDIC CCSID field on DSNTIPF. If your system uses ASCII data, then specify a mixed CCSID from Table 126 in the ASCII CODED then specify a mixed CCSID from Table 126 in the ASCII CCSID field on DSNTIPF. By specifying a CCSID for mixed data (an MCCSID), you also receive system CCSIDs for SBCS and DBCS (graphic) data. Table 125 and Table 126 show the associated CCSIDs that DB2 assigns for SBCS and DBCS data when you specify 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 125. EBCDIC double-byte coded character set identifiers (CCSIDs)

| National Language            | MCCSID | SCCSID | GCCSID | User-defined Characters |
|------------------------------|--------|--------|--------|-------------------------|
| Japanese (Extended Katakana) | 930    | 290    | 300    | 4370                    |
| Japanese (Katakana-Kanji)    | 1390   | 8482   | 16684  | 4370                    |
| Japanese (Extended Katakana) | 5026   | 290    | 4396   | 1880                    |
| Japanese (Extended English)  | 939    | 1027   | 300    | 4370                    |
| Japanese (Latin-Kanji)       | 1399   | 5123   | 16684  | 4370                    |
| Japanese (Extended English)  | 5035   | 1027   | 4396   | 1880                    |
| Korean                       | 933    | 833    | 834    | 1880                    |
| Korean                       | 1364   | 13121  | 4930   | 1880                    |
| Simplified Chinese           | 935    | 836    | 837    | 1880                    |
| Simplified Chinese           | 1388   | 13124  | 4933   | 1880                    |
| Traditional Chinese          | 937    | 28709  | 835    | 6204                    |

Table 126. ASCII double-byte coded character set identifiers (CCSIDs)

|   | National Language           | MCCSID | SCCSID | GCCSID | User-defined Characters |
|---|-----------------------------|--------|--------|--------|-------------------------|
| # | Japanese                    | 932    | 1041   | 301    | 1880                    |
|   | Japanese (Extended )        | 942    | 1041   | 301    | 1880                    |
| # | Japanese (Open environment) | 943    | 1041   | 941    | 1880                    |
|   | Japanese (HP)               | 5039   | 1041   | 1351   | 940                     |
|   | Korean                      | 949    | 1088   | 951    | 1880                    |
|   | Korean (EUC)                | 970    | 367    | 971    | 1880                    |
|   | Korean                      | 1363   | 1126   | 1362   | 1880                    |
|   | Simplified Chinese          | 1381   | 1115   | 1380   | 1880                    |
|   | Simplified Chinese (EUC)    | 1383   | 367    | 1382   |                         |



Table 126. ASCII double-byte coded character set identifiers (CCSIDs) (continued)

| National Language                  | MCCSID | SCCSID | GCCSID | User-defined Characters |
|------------------------------------|--------|--------|--------|-------------------------|
| Simplified Chinese                 | 1386   | 5210   | 1385   | 1880                    |
| Traditional Chinese                | 938    | 904    | 927    | 6204                    |
| Traditional Chinese                | 948    | 1043   | 927    | 6204                    |
| Traditional Chinese<br>(IBM Big-5) | 950    | 1114   | 947    |                         |

In Table 125 on page 526 and Table 126 on page 526, 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.

## Unicode support

Unicode is a universal encoding scheme for written characters and text that enables the exchange of data internationally. It provides a character set standard that can be used all over the world. It uses a 16-bit encoding that provides code points for more than 65,000 characters and an extension called UTF-16 that allows for encoding as many as a million more characters. It provides the ability to encode all characters used for the written languages of the world. It treats alphabetic characters, ideographic characters, and symbols equivalently because it specifies a numeric value and a name for each of its characters. It includes punctuation marks, mathematical symbols, technical symbols, geometric shapes, and dingbats. DB2 provides the following encoding forms:

- UTF-8: Unicode Transformation Format, 8-bit encoding form designed for ease of use with existing ASCII-based systems
- UTF-16: Unicode Transformation Format, 16-bit encoding form designed to provide code values for over a million characters and a superset of UCS-2. UCS-2 is Universal Character Set and is coded in 2 octets, which means that characters are represented in 16 bits per character.

**Unicode CCSIDs:** The Unicode CCSID field of panel DSNTIPF is pre-filled with 1208. DB2 chooses the CCSIDs for double-byte and single-byte values (1200 for DBCS and 367 for SBCS). CCSID 1200 corresponds to UTF-16 and CCSID 367 is for 7-bit ASCII.

## Special considerations

**Expanding Conversions:** An *expanding conversion* occurs when the length of the converted string is greater than that of the source string. For example, an expanding conversion occurs when an ASCII mixed data string that contains DBCS characters is converted to EBCDIC mixed data. Because of the addition of shift codes, an error occurs when an expanding conversion is performed on a fixed-length input host variable that requires conversion from ASCII mixed to EBCDIC mixed. The remedy is to use a varying-length string variable with a maximum length that is sufficient to contain the expansion. Expanding conversions also can occur when string data is converted to or from UNICODE.

**Contracting Conversions:** A *contracting conversion* occurs when the length of the converted string is smaller than that of the source string. For example, a contracting

conversion occurs when an EBCDIC mixed data string that contains DBCS characters is converted to ASCII mixed data due to the removal of shift codes. Contracting conversions also can occur when string data is converted to or from UNICODE.

## 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 127 on page 529 or Table 128 on page 529 for the list of ASCII and EBCDIC 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). UNICODE UTF-8 (1208) and UTF-16 (1200) support the euro symbol. UNICODE SBCS data (367) does not support the euro symbol.

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 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 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 231.
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 127. 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 128. 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.

---

## How an entry in SYSIBM.SYSSTRINGS works

The catalog table SYSIBM.SYSSTRINGS contains the following columns:

|                  |                                                   |
|------------------|---------------------------------------------------|
| <b>INCCSID</b>   | The source CCSID of a character conversion.       |
| <b>OUTCCSID</b>  | The target CCSID of a character conversion.       |
| <b>TRANSTYPE</b> | The type of conversion:                           |
| <b>SS</b>        | SBCS data to SBCS data                            |
| <b>SM</b>        | SBCS data to EBCDIC MIXED data                    |
| <b>MS</b>        | EBCDIC MIXED data to SBCS (EBCDIC and ASCII) data |
| <b>PS</b>        | ASCII MIXED data to SBCS (EBCDIC and ASCII) data  |

|           |                                              |
|-----------|----------------------------------------------|
| <b>GG</b> | GRAPHIC data to GRAPHIC data                 |
| <b>PM</b> | ASCII MIXED data to EBCDIC MIXED data        |
| <b>MM</b> | EBCDIC MIXED data to EBCDIC MIXED data.      |
| <b>MP</b> | EBCDIC MIXED to ASCII MIXED data.            |
| <b>PP</b> | ASCII MIXED to ASCII MIXED data.             |
| <b>SP</b> | SBCS (ASCII and EBCDIC) to ASCII MIXED data. |

|                  |                                                                                                                                                                                                                                                                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ERRORBYTE</b> | 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.                                                  |
| <b>SUBBYTE</b>   | 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. |
| <b>TRANSPROC</b> | 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.                                                                                  |
| <b>IBMREQD</b>   | Y indicates that the row is provided by IBM. N indicates that the row has been inserted by the user.                                                                                                                                                                                                                                            |
| <b>TRANSTAB</b>  | 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.
- If SYSTRINGS does not contain a row that describes the conversion from one CCSID to another and OS/390 V2R9 is installed, refer to OS/390 C/C++ Programming Guide for additional conversions that are supported.

## 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.

---

## Specifying locales

Rules for uppercase and lowercase usage vary according to language and country. A *locale* defines the subset of a user's environment that depends on language and cultural conventions. DB2 uses the information associated with a locale to execute UPPER, LOWER, and TRANSLATE functions in a culturally correct manner. A locale consists of two components: the first component represents a specific language and country, and the second component is a CCSID.

**Example:** In the locale, Fr\_CA.IBM-1047, Fr\_CA represents the language and country (French Canadian), and IBM-1047 is the associated CCSID.

The symbol for euro currency is supported through the modifier @EURO.

**Example:** To display results in euro dollars instead of French Francs, specify Fr\_FR@EURO.

Table 129 shows a partial list of locales supplied with OS/390 C/C++. For a more complete list of locales, see *OS/390 C/C++ Programming Guide*.

*Table 129. Examples of locales supplied with OS/390 C/C++. Excerpt of table from OS/390 C/C++ Programming Guide*

| Locale         | Language | Country     | Code set | Load module name |
|----------------|----------|-------------|----------|------------------|
| De_CH.IBM-500  | German   | Switzerland | IBM-500  | EDC\$DCEO        |
| De_CH.IBM-1047 | German   | Switzerland | IBM-1047 | EDC\$DCEY        |
| De_DE.IBM-273  | German   | Germany     | IBM-273  | EDC\$DDEB        |
| De_DE.IBM-1047 | German   | Germany     | IBM-1047 | EDC\$DDEY        |
| Fr_CA.IBM-037  | French   | Canada      | IBM-037  | EDC\$FCEA        |
| Fr_CA.IBM-1047 | French   | Canada      | IBM-1047 | EDC\$FCEY        |
| It_IT.IBM-280  | Italian  | Italy       | IBM-280  | EDC\$ITEG        |
| It_IT.IBM-1047 | Italian  | Italy       | IBM-1047 | EDC\$ITEY        |
| Ja_JP.IBM-290  | Japanese | Japan       | IBM-290  | EDC\$JAEL        |
| Ja_JP.IBM-930  | Japanese | Japan       | IBM-930  | EDC\$JAEU        |
| Ja_JP.IBM-939  | Japanese | Japan       | IBM-939  | EDC\$JAEV        |
| Ja_JP.IBM-1027 | Japanese | Japan       | IBM-1027 | EDC\$JAEX        |

## Appendix B. Directory of subsystem parameters

### Editing the subsystem parameters and DSNHDECP values

The subsystem parameter module is generated by job DSNTIJUZ each time you install, migrate, or update DB2. Seven macros expand to form this 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 data-only load module DSNHDECP is also generated by job DSNTIJUZ. It contains the application programming defaults.

### Directory of subsystem parameters and DSNHDECP values

Table 130 shows you each macro parameter, the macro where it is located, installation panel name, whether it can be updated online, and its corresponding page number. Online update capability does not apply to macro DSNHDECP so values are not listed for those parameters.

Table 130. Directory of subsystem parameters and DSNHDECP values

| Parameter  | Macro    | Panel    | Update Online | Page   |
|------------|----------|----------|---------------|--------|
| ABEXP      | DSN6SPRM | DSNTIPO  | Yes           | 153    |
| ABIND      | DSN6SPRM | DSNTIPO  | Yes           | 153    |
| AGCCSID    | DSNHDECP | DSNTIPF  | —             | 160    |
| ALCUNIT    | DSN6ARVP | DSNTIPA  | Yes           | 197    |
| ALL/dbname | DSN6SPRM | DSNTIPS  | —             | 203    |
| AMCCSID    | DSNHDECP | DSNTIPF  | —             | 160    |
| APPENSCH   | DSNHDECP | DSNTIPF  | —             | 160    |
| ARCPFX1    | DSN6ARVP | DSNTIPH  | Yes           | 103    |
| ARCPFX2    | DSN6ARVP | DSNTIPH  | Yes           | 103    |
| ARCRETN    | DSN6ARVP | DSNTIPA  | Yes           | 197    |
| ARCWRTC    | DSN6ARVP | DSNTIPA  | Yes           | 197    |
| ARCWTOR    | DSN6ARVP | DSNTIPA  | Yes           | 197    |
| ARC2FRST   | DSN6LOGP | DSNTIPO  | Yes           | 153    |
| ASCCSID    | DSNHDECP | DSNTIPF  | —             | 160    |
| ASSIST     | DSN6GRP  | DSNTIPK  | No            | 101    |
| AUDITST    | DSN6SYSP | DSNTIPN  | No            | 149    |
| AUTH       | DSN6SPRM | DSNTIPP  | No            | 184    |
| AUTHCACH   | DSN6SPRM | DSNTIPP  | Yes           | 184    |
| BACKODUR   | DSN6SYSP | DSNTIPL  | No            | 192    |
| BINDNV     | DSN6SPRM | DSNTIPP  | Yes           | 184    |
| BLKSIZE    | DSN6ARVP | DSNTIPA  | Yes           | 197    |
| BMPTOUT    | DSN6SPRM | DSNTIPI  | Yes           | 174    |
| CACHEDYN   | DSN6SPRM | DSNTIP4  | No            | 168    |
| CACHEPAC   | DSN6SPRM | DSNTIPP  | No            | 184    |
| CACHERAC   | DSN6SPRM | DSNTIPP  | No            | 184    |
| CATALOG    | DSN6ARVP | DSNTIPA  | Yes           | 197    |
| "          | DSN6SPRM | DSNTIPA2 | —             | 96     |
| CDSSRDEF   | DSN6SPRM | DSNTIP4  | Yes           | 168    |
| CHARSET    | DSNHDECP | DSNTIPF  | —             | 160    |
| CHGDC      | DSN6SPRM | DSNTIPO  | No            | 153    |
| CHKFREQ    | DSN6SYSP | DSNTIPL  | Yes           | 192    |
| CMTSTAT    | DSN6FAC  | DSNTIPR  | No            | 205    |
| COMPACT    | DSN6ARVP | DSNTIPA  | Yes           | 197    |
| COMPAT     | DSNHDECP | —        | —             | note 1 |
| CONDBAT    | DSN6SYSP | DSNTIPE  | Yes           | 138    |

Table 130. Directory of subsystem parameters and DSNHDECP values (continued)

| Parameter | Macro    | Panel    | Update Online | Page |
|-----------|----------|----------|---------------|------|
| CONTSTOR  | DSN6SPRM | DSNTIPE  | Yes           | 138  |
| COORDNTR  | DSN6GRP  | DSNTIPK  | No            | 101  |
| CTHREAD   | DSN6SYSP | DSNTIPE  | Yes           | 138  |
| DBACRVW   | DSN6SPRM | DSNTIPP  | Yes           | 184  |
| DBPROTCL  | DSN6SYSP | DSNTIP5  | Yes           | 210  |
| DATE      | DSNHDECP | DSNTIP4  | —             | 168  |
| DATELEN   | DSNHDECP | DSNTIP4  | —             | 168  |
| DDF       | DSN6FAC  | DSNTIPR  | No            | 205  |
| DEALLCT   | DSN6LOGP | DSNTIPA  | Yes           | 197  |
| DECARTH   | DSNHDECP | DSNTIPF  | —             | 160  |
| DECDIV3   | DSN6SPRM | DSNTIPF  | —             | 160  |
| DECIMAL   | DSNHDECP | DSNTIPF  | —             | 160  |
| DEFLANG   | DSNHDECP | DSNTIPF  | —             | 160  |
| DEFLTID   | DSN6SPRM | DSNTIPP  | No            | 184  |
| DELIM     | DSNHDECP | DSNTIPF  | —             | 160  |
| DESCSTAT  | DSN6SPRM | DSNTIPF  | Yes           | 160  |
| DLDFREQ   | DSN6SYSP | DSNTIPL  | Yes           | 192  |
| DLITOUT   | DSN6SPRM | DSNTIPI  | Yes           | 174  |
| DSHARE    | DSN6GRP  | DSNTIPA1 | No            | 90   |
| DSMAX     | DSN6SPRM | DSNTIPC  | Yes           | 222  |
| DSQLDELI  | DSNHDECP | DSNTIPF  | —             | 160  |
| DSSTIME   | DSN6SYSP | DSNTIPN  | Yes           | 149  |
| DYNRULES  | DSNHDECP | DSNTIPF  | —             | 160  |
| EDMBFIT   | DSN6SPRM | DSNTIP4  | —             | 168  |
| EDMDSMAX  | DSN6SPRM | DSNTIPC  | No            | 222  |
| EDMDSPAC  | DSN6SPRM | DSNTIPC  | Yes           | 222  |
| EDMPOOL   | DSN6SPRM | DSNTIPC  | Yes           | 222  |
| EDPROP    | DSN6SPRM | DSNTIPO  | No            | 153  |
| ENScheme  | DSNHDECP | DSNTIPF  | —             | 160  |
| EVALUNC   | DSN6SPRM | DSNTIP4  | Yes           | 168  |
| EXTRAREQ  | DSN6SYSP | DSNTIP5  | No            | 210  |
| EXTRASRV  | DSN6SYSP | DSNTIP5  | No            | 210  |
| EXTSEC    | DSN6SYSP | DSNTIPR  | No            | 205  |
| GCCSID    | DSNHDECP | DSNTIPF  | —             | 160  |
| GRPNAME   | DSN6GRP  | DSNTIPK  | No            | 101  |
| HOPAUTH   | DSN6SPRM | DSNTIP5  | No            | 210  |
| IDBACK    | DSN6SYSP | DSNTIPE  | Yes           | 138  |
| IDFORE    | DSN6SYSP | DSNTIPE  | Yes           | 138  |
| IDTHTOIN  | DSN6FAC  | DSNTIPR  | No            | 205  |
| IDXBPOOL  | DSN6SYSP | DSNTIP1  | Yes           | 143  |
| IMMEDWRI  | DSN6GRP  | DSNTIP4  | No            | 168  |
| IRLMAUT   | DSN6SPRM | DSNTIPI  | No            | 174  |
| IRLMPRC   | DSN6SPRM | DSNTIPI  | No            | 174  |
| IRLMRWT   | DSN6SPRM | DSNTIPI  | No            | 174  |
| IRLMSID   | DSN6SPRM | DSNTIPI  | No            | 174  |
| IRLMSWT   | DSN6SPRM | DSNTIPI  | Yes           | 174  |
| LBACKOUT  | DSN6SYSP | DSNTIPL  | No            | 192  |
| LC_CTYPE  | DSNHDECP | DSNTIPF  | —             | 160  |
| LEMAX     | DSN6SPRM | DSNTIP7  | No            | 136  |
| LOBVALA   | DSN6SYSP | DSNTIP7  | No            | 136  |
| LOBVALS   | DSN6SYSP | DSNTIP7  | No            | 174  |
| LOGAPSTG  | DSN6SYSP | DSNTIPL  | No            | 192  |
| MAXARCH   | DSN6LOGP | DSNTIPA  | No            | 197  |
| MAXDBAT   | DSN6SYSP | DSNTIPE  | Yes           | 138  |
| MAXKEEPD  | DSN6SPRM | DSNTIPE  | No            | 138  |
| MAXRBLK   | DSN6SPRM | DSNTIPC  | Yes           | 222  |
| MAXRTU    | DSN6LOGP | DSNTIPA  | Yes           | 197  |



Table 130. Directory of subsystem parameters and DSNHDECP values (continued)

| Parameter     | Macro    | Panel   | Update Online | Page   |
|---------------|----------|---------|---------------|--------|
| MAXTYPE1      | DSN6FAC  | DSNTIPR | No            | 205    |
| MCCSID        | DSNHDECP | DSNTIPF | —             | 160    |
| MEMBNAME      | DSN6GRP  | DSNTIPK | No            | 101    |
| MINSTOR       | DSN6SPRM | DSNTIPE | Yes           | 138    |
| MIXED         | DSNHDECP | DSNTIPF | —             | 160    |
| MON           | DSN6SYSP | DSNTIPN | No            | 149    |
| MONSIZE       | DSN6SYSP | DSNTIPN | No            | 149    |
| NPGTHRS       | DSN6SPRM | —       | —             | 242    |
| NUMLKTS       | DSN6SPRM | DSNTIPJ | Yes           | 179    |
| NUMLKUS       | DSN6SPRM | DSNTIPJ | Yes           | 179    |
| OJPERFEH      | DSN6SPRM | —       | No            | 242    |
| OPTHINTS      | DSN6SPRM | DSNTIP4 | Yes           | 168    |
| OPTOPSE       | DSN6SPRM | —       | No            | 242    |
| OUTBUFF       | DSN6LOGP | DSNTIPL | No            | 192    |
| PARAMDEG      | DSN6SPRM | DSNTIP4 | —             | 168    |
| PARTKEYU      | DSN6SPRM | DSNTIP4 | —             | 168    |
| PCLOSEN       | DSN6SYSP | DSNTIPL | Yes           | 192    |
| PCLOSET       | DSN6SYSP | DSNTIPL | Yes           | 192    |
| POOLINAC      | DSN6FAC  | DSNTIP5 | No            | 210    |
| PRIQTY        | DSN6ARVP | DSNTIPA | Yes           | 197    |
| PROTECT       | DSN6ARVP | DSNTIPP | Yes           | 184    |
| PTASKROL      | DSN6SYSP | —       | Yes           | 242    |
| QUIESCE       | DSN6ARVP | DSNTIPA | Yes           | 197    |
| RECALL        | DSN6SPRM | DSNTIPO | No            | 153    |
| RECALLD       | DSN6SPRM | DSNTIPO | Yes           | 153    |
| RELCURHL      | DSN6SPRM | DSNTIP4 | Yes           | 168    |
| RESTART/DEFER | DSN6SPRM | DSNTIPS | —             | 203    |
| RESYNC        | DSN6FAC  | DSNTIPR | No            | 205    |
| RETLWAIT      | DSN6SPRM | DSNTIPI | Yes           | 174    |
| RETVLCFK      | DSN6SPRM | DSNTIP4 | Yes           | 168    |
| RGFCOLID      | DSN6SPRM | DSNTIPZ | No            | 216    |
| RGFDBNAM      | DSN6SPRM | DSNTIPZ | No            | 216    |
| RGFDEDPL      | DSN6SPRM | DSNTIPZ | No            | 216    |
| RGFDEFLT      | DSN6SPRM | DSNTIPZ | No            | 216    |
| RGFESCP       | DSN6SPRM | DSNTIPZ | No            | 216    |
| RGFFULLQ      | DSN6SPRM | DSNTIPZ | No            | 216    |
| RGFINSTL      | DSN6SPRM | DSNTIPZ | No            | 216    |
| RGFNMORT      | DSN6SPRM | DSNTIPZ | No            | 216    |
| RGFNMPRT      | DSN6SPRM | DSNTIPZ | No            | 216    |
| RLF           | DSN6SYSP | DSNTIPO | No            | 153    |
| RLFAUTH       | DSN6SYSP | DSNTIPP | Yes           | 184    |
| RLFERR        | DSN6SYSP | DSNTIPO | Yes           | 153    |
| RLFERRD       | DSN6FAC  | DSNTIPR | Yes           | 205    |
| RLFTBL        | DSN6SYSP | DSNTIPO | Yes           | 153    |
| ROUTCDE       | DSN6SYSP | DSNTIPO | No            | 153    |
| RRULOCK       | DSN6SPRM | DSNTIPI | Yes           | 174    |
| SCCSID        | DSNHDECP | DSNTIPF | —             | 160    |
| SECQTY        | DSN6ARVP | DSNTIPA | Yes           | 197    |
| SEQCACH       | DSN6SPRM | DSNTIPE | Yes           | 138    |
| SEQPRES       | DSN6SPRM | DSNTIPE | Yes           | 138    |
| SITETYP       | DSN6SPRM | DSNTIPO | No            | 153    |
| SMFACCT       | DSN6SYSP | DSNTIPN | No            | 149    |
| SMFSTAT       | DSN6SYSP | DSNTIPN | No            | 149    |
| SMSDCFL       | DSN6SPRM | —       | —             | 242    |
| SMSDCIX       | DSN6SPRM | —       | —             | 242    |
| SPRMLTD       | DSN6SPRC | —       | —             | note 2 |
| SQLDELI       | DSNHDECP | DSNTIPF | —             | 160    |

Table 130. Directory of subsystem parameters and DSNHDECP values (continued)

| Parameter | Macro    | Panel   | Update Online | Page |
|-----------|----------|---------|---------------|------|
| SRTPOOL   | DSN6SPRM | DSNTIPC | No            | 222  |
| SSID      | DSNHDECP | DSNTIPM | —             | 188  |
| STARJOIN  | DSN6SPRM | —       | No            | 242  |
| STATHIST  | DSN6SPRM | DSNTIPO | No            | 153  |
| STATIME   | DSN6SYSP | DSNTIPN | Yes           | 149  |
| STATROLL  | DSN6SPRM | DSNTIPO | Yes           | 153  |
| STATSINT  | DSN6SPRM | DSNTIPO | Yes           | 153  |
| STDSQL    | DSNHDECP | DSNTIP4 | —             | 168  |
| STORMXAB  | DSN6SYSP | DSNTIPX | Yes           | 214  |
| STORPROC  | DSN6SYSP | DSNTIPX | No            | 214  |
| STORTIME  | DSN6SYSP | DSNTIPX | Yes           | 214  |
| SUPERRS   | DSN6SPRM | DSNTIPM | Yes           | 188  |
| SYNCVAL   | DSN6SYSP | DSNTIPN | Yes           | 149  |
| SYSADM    | DSN6SPRM | DSNTIPP | No            | 184  |
| SYSADM2   | DSN6SPRM | DSNTIPP | No            | 184  |
| SYSOPR1   | DSN6SPRM | DSNTIPP | No            | 184  |
| SYSOPR2   | DSN6SPRM | DSNTIPP | No            | 184  |
| TBSBPOOL  | DSN6SYSP | DSNTIP1 | Yes           | 143  |
| TCPALVER  | DSN6FAC  | DSNTIP5 | No            | 210  |
| TCPKPALV  | DSN6FAC  | DSNTIP5 | No            | 210  |
| TIME      | DSNHDECP | DSNTIP4 | —             | 168  |
| TIMELEN   | DSNHDECP | DSNTIP4 | —             | 168  |
| TRACSTR   | DSN6SYSP | DSNTIPN | No            | 149  |
| TRACTBL   | DSN6SYSP | DSNTIPN | No            | 149  |
| TRKRSITE  | DSN6SPRM | DSNTIPO | No            | 153  |
| TSTAMP    | DSN6ARVP | DSNTIPH | Yes           | 103  |
| TWOACTV   | DSN6LOGP | DSNTIPH | No            | 103  |
| TWOARCH   | DSN6LOGP | DSNTIPH | No            | 103  |
| TWOBSDS   | DSN6LOGP | —       | No            | 294  |
| UGCCSID   | DSNHDECP | DSNTIPF | —             | 160  |
| UMCCSID   | DSNHDECP | DSNTIPF | —             | 160  |
| UNIT      | DSN6ARVP | DSNTIPA | Yes           | 197  |
| UNIT2     | DSN6ARVP | DSNTIPA | Yes           | 197  |
| URCHKTH   | DSN6SYSP | DSNTIPL | Yes           | 192  |
| URLGWTH   | DSN6SYSP | DSNTIPL | Yes           | 192  |
| USCCSID   | DSNHDECP | DSNTIPF | —             | 160  |
| UTIMOUT   | DSN6SPRM | DSNTIPI | Yes           | 174  |
| WLMENV    | DSN6SYSP | DSNTIPX | Yes           | 214  |
| XLKUPDLT  | DSN6SPRM | DSNTIPI | No            | 174  |

Note 1: Serviceability parameter

Note 2: SPRMLTD is introduced in macro member DSN6SPRC of the *prefix.SDSNMACS* data set. The default value is 10. The SPRMLTD value multiplied by 32K is used by DB2 to determine a size threshold for compression of the in-memory only copy of the Database Descriptor (DBD) for the new TEMP database when the database is no longer in use. This parameter has no effect on the actual size of the DB2 Directory copy of the DBD on DASD. You can use the new declared temporary tables without editing or changing the SPRMLTD value. You must reassemble and re-link the DB2 initialization parameter module, DSNZPARM, for each DB2 subsystem or Data Sharing member that will have the new TEMP database.

## Appendix C. DB2 utilities packaging

DB2 Version 7 provides utilities in various packages. Core utilities are included with DB2. Other utilities are available in three separate products. All DB2 utilities operate on catalog, directory and sample objects: without requiring any additional products.

The core utilities include:

- CATMAINT
- DIAGNOSE
- LISTDEF
- OPTIONS
- QUIESCE
- REPAIR
- REPORT
- TEMPLATE
- all standalone utilities

The DB2 utility products are:

- 5655-E63: DB2 Operational Utilities (FMID JDB771K)
  - COPY
  - EXEC SQL
  - LOAD
  - REBUILD INDEX
  - RECOVER
  - REORG INDEX
  - REORG TABLESPACE
  - RUNSTATS
  - STOSPACE
  - UNLOAD
- 5655-E62: DB2 Diagnostic and Recovery Utilities (FMID JDB771M)
  - CHECK DATA
  - CHECK INDEX
  - CHECK LOB
  - COPY
  - COPYTOCOPY
  - MERGECOPY
  - MODIFY RECOVERY
  - MODIFY STATISTICS
  - REBUILD INDEX
  - RECOVER
- 5697-E98: DB2 Utilities Suite (FMIDs JDB771K and JDB771M)
  - CHECK DATA
  - CHECK INDEX
  - CHECK LOB
  - COPY
  - COPYTOCOPY
  - EXEC SQL
  - LOAD
  - MERGECOPY
  - MODIFY RECOVERY
  - MODIFY STATISTICS
  - REBUILD INDEX
  - RECOVER
  - REORG INDEX

- REORG TABLESPACE
- RUNSTATS
- STOSPACE
- UNLOAD

---

## SMP/E jobs for DB2 utility products

To load the DB2 utility products, use System Modification Program Extended (SMP/E). SMP/E processes the installation tapes or cartridges and creates DB2 distribution target libraries.

Several jobs are provided that invoke SMP/E. These jobs are on the tape or cartridge you received with the utility product. The job prologues in these jobs contain directions on how to tailor the job for your site. Follow these directions carefully to ensure that your DB2 for OS/390 and z/OS SMP/E process functions correctly. To copy the jobs from the tapes submit the copy job listed in the DB2 Program Directory.

## Installing DB2 Operational Utilities

The SMP/E RECEIVE job, DSNRECVK, loads the DB2 Operational Utilities program modules, macros, and procedures into temporary data sets (SMPTLIBs). If this job fails or abends, correct the problem and rerun the job. Use “SMP/E step 4: Run the receive jobs: DSNRECV1, DSNRECV2, DSNRECV3, DSNRECV4” on page 64 as a guide to help you with this job.

The SMP/E APPLY job, DSNAPPLK, copies and link-edits the DB2 program modules, macros, and procedures into the DB2 target libraries. Use “SMP/E step 6: Run the apply jobs: DSNAPPL1, DSNAPPL2” on page 65 as a guide to help you with this job.

The SMP/E ACCEPT job, DSNACCPK, copies the DB2 Operational Utility program modules, macros, and procedures into the DB2 distribution libraries. Use “SMP/E step 7: Run the accept jobs: DSNACEP1, DSNACEP2” on page 66 as a guide to help you with this job.

## Installing DB2 Diagnostic & Recovery Utilities

The SMP/E RECEIVE job, DSNRECVK, loads the DB2 Diagnostic and Recovery Utilities program modules, macros, and procedures into temporary data sets (SMPTLIBs). If this job fails or abends, correct the problem and rerun the job. Use “SMP/E step 4: Run the receive jobs: DSNRECV1, DSNRECV2, DSNRECV3, DSNRECV4” on page 64 as a guide to help you with this job.

The SMP/E APPLY job, DSNAPPLM, copies and link-edits the DB2 Diagnostic and Recovery Utilities program modules, macros, and procedures into the DB2 target libraries. Use “SMP/E step 6: Run the apply jobs: DSNAPPL1, DSNAPPL2” on page 65 as a guide to help you with this job.

The SMP/E ACCEPT job, DSNACCPM, copies the DB2 Diagnostic and Recovery Utilities program modules, macros, and procedures into the DB2 distribution libraries. Use “SMP/E step 7: Run the accept jobs: DSNACEP1, DSNACEP2” on page 66 as a guide to help you with this job.

## Installing DB2 Utilities Suite

The SMP/E RECEIVE job, DSNRECVS, loads the DB2 Diagnostic and Recovery Utilities program modules, macros, and procedures into temporary data sets (SMPTLIBS). The SMP/E RECEIVE job, DSNRECVK, loads the DB2 Operational Utilities program modules, macros, and procedures into temporary data sets (SMPTLIBS). If these jobs fail or abend, correct the problem and rerun the jobs. Use “SMP/E step 4: Run the receive jobs: DSNRECV1, DSNRECV2, DSNRECV3, DSNRECV4” on page 64 as a guide to help you with the RECEIVE job.

The SMP/E APPLY job, DSNAPPLS, copies and link-edits both the DB2 Diagnostic and Recovery Utilities program modules, macros, and procedures; and the the DB2 Operational Utilities program modules, macros, and procedures into the DB2 target libraries. Use “SMP/E step 6: Run the apply jobs: DSNAPPL1, DSNAPPL2” on page 65 as a guide to help you with the APPLY job.

The SMP/E ACCEPT job, DSNACCPs, copies the DB2 Diagnostic and Recovery Utilities program modules, macros, and procedures; and the DB2 Operational Utility program modules, macros, and procedures into the DB2 distribution libraries. Use “SMP/E step 7: Run the accept jobs: DSNACEP1, DSNACEP2” on page 66 as a guide to help you with the ACCEPT job.

---

## How to operate in a mixed-release data sharing environment

The utilities batch module DSNUTILB, is split into multiple parts: a release-independent module called DSNUTILB and multiple release-dependent modules, DSNUT710 and the utility-dependent load modules listed in Table 131 . To operate in a mixed-release data sharing environment, you must have the release-dependent modules from both releases and all applicable utility-dependent modules available to the utility jobs that operate across the data sharing group. The procedure for sharing utility modules is explained in Chapter 4 of *DB2 Data Sharing: Planning and Administration*. Use the information in Table 131 and the procedures outlined in Chapter 4 of *DB2 Data Sharing: Planning and Administration* to implement a mixed-release data sharing environment.

With Version 7, there are separate load modules and aliases for each utility. For a cross reference between the utility name, alias name and load module names, see Table 131. Each utility product contains unique load module names and some utilities are available in multiple utility products. Therefore some utilities have multiple load module names because the utility ships with different products. The core utility load modules use DSNU7CLx naming convention. The Operational utilities use DSNU7OLx names. The Diagnostic and Recovery utilities use DSNU7RLx names.

Table 131. Relationship between utility names, aliases and utility load modules

| Utility name | Alias name | Load module name   |
|--------------|------------|--------------------|
| CATMAINT     | DSNU71AA   | DSBU7CLA           |
| CHECK        | DSNU71AB   | DSNU7RLB           |
| COPY         | DSNU71AC   | DSNU7OLC, DSNU7RLC |
| COPYTOCOPY   | DSNU71AT   | DSNU7RLT           |
| DIAGNOSE     | DSNU71AD   | DSNU7CLD           |
| EXEC SQL     | DSNU71AU   | DSNU7OLU           |
| LISTDEF      | DSNU71AE   | DSNU7CLE           |

Table 131. Relationship between utility names, aliases and utility load modules (continued)

| Utility name                             | Alias name | Load module name   |
|------------------------------------------|------------|--------------------|
| LOAD                                     | DSNU71AF   | DSNU7OLF           |
| MERGECOPY                                | DSNU71AG   | DSNU7RLG           |
| MODIFY RECOVERY and<br>MODIFY STATISTICS | DSNU71AH   | DSNU7RLH           |
| OPTIONS                                  | DSNU71AI   | DSNU7CLI           |
| QUIESCE                                  | DSNU71AJ   | DSNU7CLJ           |
| REBUILD INDEX                            | DSNU71AK   | DSNU7OLK, DSNU7RLK |
| RECOVER                                  | DSNU71AL   | DSNU7OLL, DSNU7RLL |
| REORG INDEX and REORG<br>TABLESPACE      | DSNU71AM   | DSNU7OLM           |
| REPAIR                                   | DSNU71AN   | DSNU7CLN           |
| REPORT                                   | DSNU71AO   | DSNU7CLO           |
| RUNSTATS                                 | DSNU71AP   | DSNU7OLP           |
| STOSPACE                                 | DSNU71AQ   | DSNU7OLQ           |
| TEMPLATE                                 | DSNU71AR   | DSNU7CLR           |
| UNLOAD                                   | DSNU71AS   | DSNU7OLS           |

## Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

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

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106-0032, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs

and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

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

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

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

---

## Programming interface information

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

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

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

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 International Business Machines Corporation in the United States, or other countries, or both:

|                                              |                      |
|----------------------------------------------|----------------------|
| 3090                                         | ESA/390              |
| AD/Cycle                                     | ES/9000              |
| AIX                                          | GDDM                 |
| APL2                                         | Hiperspace           |
| AS/400                                       | IBM                  |
| BookManager                                  | IBMLink              |
| CICS                                         | IMS                  |
| CICS/ESA                                     | IMS/ESA              |
| CICS/MVS                                     | Language Environment |
| COBOL/370                                    | MVS/DFP              |
| C/MVS                                        | MVS/ESA              |
| C/370                                        | MVS/SP               |
| DATABASE 2                                   | MVS/XA               |
| DataHub                                      | Net.Data             |
| DataPropagator                               | OpenEdition          |
| DB2                                          | OS/2                 |
| DB2 Connect                                  | OS/390               |
| DB2 Universal Database                       | OS/400               |
| DFSMS                                        | Parallel Sysplex     |
| DFSMSdfp                                     | PR/SM                |
| DFSMSdss                                     | QMF                  |
| DFSMShsm                                     | RACF                 |
| DFSMS/MVS                                    | RAMAC                |
| DFSORT                                       | RMF                  |
| Distributed Relational Database Architecture | SOMobjects           |
| DRDA                                         | SQL/DS               |
| eNetwork                                     | System/370           |
| Enterprise System/3090                       | System/390           |
| Enterprise System/9000                       | VTAM                 |

Tivoli and NetView are trademarks of Tivoli Systems Inc. in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

---

# Glossary

The following terms and abbreviations are defined as they are used in the DB2 library.

## 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 *DB2 Messages and Codes*.

**abnormal end of task (abend).** Termination of a task, job, or subsystem because of an error condition that recovery facilities cannot resolve during execution.

**access method services.** The facility that is used to define and reproduce VSAM key-sequenced data sets.

**access path.** The path that is used to locate data that is specified in SQL statements. An access path can be indexed or sequential.

**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 fit on the active log.

**address space.** A range of virtual storage pages that is identified by a number (ASID) and a collection of segment and page tables that 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 that contains a task that is connected to DB2 has exactly one address space connection, even though more than one task control block (TCB) can be present. See also *allied address space* and *task control block*.

**agent.** As used in DB2, the structure that associates all processes that are involved in a DB2 unit of work. An *allied agent* is generally synonymous with an *allied thread*. *System agents* are units of work that process independently of the allied agent, such as prefetch processing, deferred writes, and service tasks.

**alias.** An alternative 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 that is external to DB2 and that is connected to DB2. An allied address space is capable of requesting DB2 services.

**allied thread.** A thread that originates at the local DB2 subsystem and that can access data at a remote DB2 subsystem.

**already verified.** An LU 6.2 security option that allows DB2 to provide the user's verified authorization ID when allocating a conversation. The user is not validated by the partner DB2 subsystem.

**ambiguous cursor.** A database cursor that is not defined with the FOR FETCH ONLY clause or the FOR UPDATE OF clause, 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 either PREPARE or EXECUTE IMMEDIATE SQL statements.

**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 that is used to define DB2 to VTAM as an application program that uses SNA LU 6.2 protocols.

**application.** A program or set of programs that performs a task; for example, a payroll application.

| **application-directed connection.** A connection that  
| an application manages using the SQL CONNECT  
| statement.

**application plan.** The control structure that is produced during the bind process. DB2 uses the application plan to process SQL statements that it encounters 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 programming interface (API).** A functional interface that is supplied by the operating system or by a separately orderable licensed program that allows an application program that is written in a high-level language to use specific data or functions of the operating system or licensed program.

| **application requester.** The component on a remote  
| system that generates DRDA requests for data on  
| behalf of an application. An application requester  
| accesses a DB2 database server using the DRDA  
| application-directed protocol.

| **application server.** The target of a request from a  
| remote application. In the DB2 environment, the

## archive log • call attachment facility (CAF)

| application server function is provided by the distributed  
| data facility and is used to access DB2 data from  
| remote applications.

**archive log.** The portion of the DB2 log that contains log records that have been copied from the active log.

| **ASCII.** An encoding scheme that is used to represent  
| strings in many environments, typically on PCs and  
| workstations. Contrast with *EBCDIC* and *Unicode*.

**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 is 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 that is caused by a suspected defect in a current release of an IBM licensed program.

**authorized program facility (APF).** A facility that permits the identification of programs that are authorized to use restricted functions.

**auxiliary index.** An index on an auxiliary table in which each index entry refers to a LOB.

**auxiliary table.** A table that stores columns outside the table in which they are defined. Contrast with *base table*.

## 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.** (1) A table that is created by the SQL CREATE TABLE statement and that holds persistent data. Contrast with *result table* and *temporary table*.

(2) A table containing a LOB column definition. The actual LOB column data is not stored with the base table. The base table contains a row identifier for each row and an indicator column for each of its LOB columns. Contrast with *auxiliary 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.

**binary large object (BLOB).** A sequence of bytes where the size of the value ranges from 0 bytes to 2 GB–1. Such a string does not have an associated CCSID.

**bind.** The process by which the output from the SQL precompiler is converted to a usable control structure, often called an access plan, application plan, or package. During this process, access paths to the data are selected and some authorization checking is performed. The types of bind are:

**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.

**BLOB.** Binary large object.

**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 that is reserved to satisfy the buffering requirements for one or more table spaces or indexes.

**built-in function.** A function that DB2 supplies. Contrast with *user-defined function*.

## C

**CAF.** Call attachment facility.

**call attachment facility (CAF).** A DB2 attachment facility for application programs that run in TSO or MVS batch. The CAF is an alternative to the DSN command processor and provides greater control over the execution environment.

**cascade delete.** The way in which DB2 enforces referential constraints when it deletes all descendent rows of a deleted parent row.

**cast function.** A function that is used to convert instances of a (source) data type into instances of a different (target) data type. In general, a cast function has the name of the target data type. It has one single argument whose type is the source data type; its return type is the target data type.

**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.** Communications database.

**character large object (CLOB).** A sequence of bytes representing single-byte characters or a mixture of single- and double-byte characters where the size of the value can be up to 2 GB–1. In general, character large object values are used whenever a character string might exceed the limits of the VARCHAR type.

**character set.** A defined set of characters.

| **character string.** A sequence of bytes that represent  
| bit data, single-byte characters, or a mixture of  
| single-byte and multibyte characters.

**check constraint.** See *table check constraint*.

**check integrity.** The condition that exists when each row in a table conforms to the table check constraints that are defined on that table. Maintaining check integrity requires DB2 to enforce 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 of rows that violate referential constraints, table check constraints, or both.

**checkpoint.** A point at which DB2 records internal status information on the DB2 log; the recovery process uses this information if DB2 abnormally terminates.

**CI.** Control interval.

**CICS.** Represents (in this publication) one of the following products:

**CICS Transaction Server for OS/390:** Customer Information Control System Transaction Server for OS/390

**CICS/ESA:** Customer Information Control System/Enterprise Systems Architecture

**CICS/MVS:** Customer Information Control System/Multiple Virtual Storage

**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.** A notification to DB2 that an object is being accessed. Claims prevent drains from occurring until the claim is released, which usually occurs at a commit point. Contrast with *drain*.

**claim class.** A specific type of object access that 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.

**CLOB.** Character large object.

**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.

**coexistence.** During migration, the period of time in which two releases exist in the same data sharing group.

**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 operation that derives its result by using values from one or more rows. Contrast with *scalar function*.

## "come from" checking • created temporary table

**"come from" checking.** An LU 6.2 security option that 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. A command is 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 that are 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 multisite 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 that are 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.** A token (such as =, >, <) that is 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 that is supplied by the attachment facility and that is associated with a specific address space connection.

**consistency token.** A timestamp that is 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*, *table check constraint*, and *uniqueness 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 4 bytes at the end of each control interval; it describes the free space, if any, in the control interval.

**conversation.** Communication, which is based on LU 6.2 or Advanced Program-to-Program Communication (APPC), between an application and a remote transaction program over an SNA logical unit-to-logical unit (LU-LU) session that allows communication while processing a transaction.

**coordinator.** The system component that coordinates the commit or rollback of a unit of work that includes work that is done on one or more other systems.

**correlated subquery.** A subquery (part of a WHERE or HAVING clause) that is applied to a row or group of rows of a table or view that is named in an outer subselect statement.

**correlation ID.** An identifier that is 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.

**C++ member.** A data object or function in a structure, union, or class.

**C++ member function.** An operator or function that is declared as a member of a class. A member function has access to the private and protected data members and to the member functions of objects in its class. Member functions are also called methods.

**C++ object.** (1) A region of storage. An object is created when a variable is defined or a new function is invoked. (2) An instance of a class.

**CRC.** Command recognition character.

**CRCR.** Conditional restart control record. See also *conditional restart*.

**create link pack area (CLPA).** An option used during IPL to initialize the link pack pageable area.

**created temporary table.** A table that holds temporary data and is defined with the SQL statement CREATE GLOBAL TEMPORARY TABLE. Information about created temporary tables is stored in the DB2 catalog, so this kind of table is persistent and can be shared

across application processes. Contrast with *declared temporary table*. See also *temporary table*.

**cross-memory linkage.** A method for invoking a program in a different address space. The invocation is synchronous with respect to the caller.

**CS.** Cursor stability.

**CSA.** Common service area.

**CT.** Cursor table.

**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 that an application program uses 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 copy of the skeleton cursor table that is 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 table is a descendent of the last table. 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 that accesses data at the local subsystem on behalf of a remote subsystem.

**database administrator (DBA).** An individual who is responsible for designing, developing, operating, safeguarding, maintaining, and using a database.

**database descriptor (DBD).** An internal representation of a DB2 database definition, which reflects the data definition that is in the DB2 catalog. The objects that are 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 the access to the data stored within it.

## cross-memory linkage • DB2 command

**database request module (DBRM).** A data set member that is created by the DB2 precompiler and that contains information about SQL statements. DBRMs are used in the bind process.

| **database server.** The target of a request from a local  
| application or an intermediate database server. In the  
| DB2 environment, the database server function is  
| provided by the distributed data facility to access DB2  
| data from local applications, or from a remote database  
| server that acts as an intermediate database server.

**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 (ddname).** 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 space.** A range of up to 2 GB of contiguous virtual storage addresses that a program can directly manipulate. Unlike an address space, a data space can hold only data; it does not contain common areas, system data, or programs.

**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.

**DBCLOB.** Double-byte character large object.

**DBCS.** Double-byte character set.

**DBD.** Database descriptor.

**DBID.** Database identifier.

**DBMS.** Database management system.

**DBRM.** Database request module.

**DB2 catalog.** Tables that are maintained by DB2 and 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

## DB2 for VSE & VM • double-byte character set (DBCS)

information on current users, to start or stop databases, to display information on the status of databases, and so on.

**DB2 for VSE & VM.** The IBM DB2 relational database management system for the VSE and VM operating systems.

**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.

**DCLGEN.** Declarations generator.

**DDF.** Distributed data facility.

**ddname.** 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.

**declared temporary table.** A table that holds temporary data and is defined with the SQL statement DECLARE GLOBAL TEMPORARY TABLE. Information about declared temporary tables is not stored in the DB2 catalog, so this kind of table is not persistent and can only be used by the application process that issued the DECLARE statement. Contrast with *created temporary table*. See also *temporary table*.

**default value.** A predetermined value, attribute, or option that is assumed when no other is explicitly specified.

**deferred write.** The process of asynchronously writing changed data pages to disk.

**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) that 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 a dependent in at least one referential constraint.

**descendent.** An object that is a dependent of an object or is the dependent of a descendent of an object.

**descendent row.** A row that is dependent on another row, or a row that is a descendent of a dependent row.

**descendent table.** A table that is a dependent of another table, or a table that is a descendent of a dependent table.

**DFHSM.** Data Facility Hierarchical Storage Manager.

**DFP.** Data Facility Product (in MVS).

**dimension.** A data category such as time, products, or markets. The elements of a dimension are referred to as members. Dimensions offer a very concise, intuitive way of organizing and selecting data for retrieval, exploration, and analysis. See also *dimension table*.

**dimension table.** The representation of a dimension in a star schema. Each row in a dimension table represents all of the attributes for a particular member of the dimension. See also *dimension*, *star schema*, and *star join*.

**direct access storage device (DASD).** A device in which access time is independent of the location of the data.

**directory.** The DB2 system database that contains internal objects such as database descriptors and skeleton cursor tables.

**Distributed Computing Environment MVS/ESA (DCE MVS/ESA).** A set of technologies that are provided by the Open Software Foundation to implement distributed computing.

**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.

**double-byte character large object (DBCLOB).** A sequence of bytes representing double-byte characters where the size of the values can be up to 2 GB. In general, double-byte character large object values are used whenever a double-byte character string might exceed the limits of the VARGRAPHIC type.

**double-byte character set (DBCS).** A set of characters, which are used by national languages such as Japanese and Chinese, that have more symbols



than can be represented by a single byte. Each character is 2 bytes in length. Contrast with *single-byte character set* and *multibyte character set*.

**drain.** The act of acquiring a locked resource by quiescing access to that object.

**drain lock.** A lock on a claim class that prevents a claim from occurring.

**DRDA.** Distributed Relational Database Architecture.

**DRDA access.** An open method of accessing distributed data that you can use to connect to another database server to execute packages that were previously bound at the server location. You use the SQL CONNECT statement or an SQL statement with a three-part name to identify the server. Contrast with *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

**EA-enabled table space.** A table space or index space that is enabled for extended addressability and that contains individual partitions (or pieces, for LOB table spaces) that are greater than 4 GB.

**EBCDIC.** Extended binary coded decimal interchange code. An encoding scheme that is used to represent character data in the OS/390, MVS, VM, VSE, and OS/400® environments. Contrast with *ASCII* and *Unicode*.

**EDM pool.** A pool of main storage that is used for database descriptors, application plans, authorization cache, application packages, and dynamic statement caching.

**EID.** Event identifier.

**embedded SQL.** SQL statements that are coded within an application program. See *static SQL*.

**EOM.** End of memory.

**EOT.** End of task.

**equijoin.** A join operation in which the join-condition has the form *expression = expression*.

**error page range.** A range of pages that are considered to be physically damaged. DB2 does not allow users to access any pages that fall within this range.

**ESDS.** Entry sequenced data set.

**ESMT.** External subsystem module table (in IMS).

**EUR.** IBM European Standards.

**exception table.** A table that holds rows that violate referential constraints or table check constraints that the CHECK DATA utility finds.

**exclusive lock.** A lock that prevents concurrently executing application processes from reading or changing data. Contrast with *share 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.

**extended recovery facility (XRF).** A facility that minimizes the effect of failures in MVS, VTAM, the host processor, or high-availability applications during sessions between high-availability applications and designated terminals. This facility provides an alternative subsystem to take over sessions from the failing subsystem.

**external function.** A function for which the body is written in a programming language that takes scalar argument values and produces a scalar result for each invocation. Contrast with *sourced function*, *built-in function*, and *SQL function*.

**External subsystem module table (ESMT).** The name of the external subsystem module table, which specifies which attachment modules must be loaded by IMS.

## 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 that is 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 • image copy

**foreign key.** A column or set of columns in a dependent table of a constraint relationship. The key must have the same number of columns, with the same descriptions, as the primary key of the parent table. Each foreign key value must either match a parent key value in the related parent table or be null.

**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 amount of unused space in a page; that is, the space that is not used to store records or control information is free space.

**full outer join.** The result of a join operation that includes the matched rows of both tables that are being joined and preserves the unmatched rows of both tables. See also *join*.

**function.** A mapping, embodied as a program (the function body), invocable by means of zero or more input values (arguments), to a single value (the result). See also *column function* and *scalar function*.

Functions can be user-defined, built-in, or generated by DB2. (See *built-in function*, *cast function*, *external function*, *sourced function*, *SQL function*, and *user-defined function*.)

## G

**GB.** Gigabyte (1 073 741 824 bytes).

**GBP.** Group buffer pool.

**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 (GBP).** A coupling facility cache structure that is 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.

**hiperspace.** A range of up to 2 GB of contiguous virtual storage addresses that a program can use as a buffer. Like a data space, a hiperspace can hold user data; it does not contain common areas or system data. Unlike an address space or a data space, data in a hiperspace is not directly addressable. To manipulate data in a hiperspace, bring the data into the address space in 4-KB blocks.

**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 that is written in a host language and that contains embedded SQL statements.

**host structure.** In an application program, a structure that is referenced by embedded SQL statements.

**host variable.** In an application program, an application variable that is referenced by embedded SQL statements.

**HSM.** Hierarchical storage manager.

## I

**ICF.** Integrated catalog facility.

**IDCAMS.** An IBM program that is used to process access method services 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 that is contained in the access method services catalog.

**identify.** A request that an attachment service program in an address space that is separate from DB2 issues via the MVS subsystem interface to inform DB2 of its existence and to 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.** 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 continues 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) that 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 that is used to determine the order of index entries.

**index space.** A page set that is used to store the entries of one index.

**indicator variable.** A variable that is 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 an individual unit of recovery is to be committed or rolled back. At emergency restart, if DB2 lacks the information it needs to make this decision, the status of the unit of recovery is *indoubt* until DB2 obtains this information from the coordinator. More than one unit of recovery can be indoubt at restart.

**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 at restart. 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.

**instrumentation facility interface (IFI).** A programming interface that enables programs to obtain online trace data about DB2, to submit DB2 commands, and to pass data to DB2.

**Interactive System Productivity Facility (ISPF).** An IBM licensed program that provides interactive dialog services.

**intermediate database server.** The target of a request from a local application or a remote application requester that is forwarded to another database server. In the DB2 environment, the remote request is forwarded transparently to another database server if the object that is referenced by a three-part name does not reference the local location.

**internal resource lock manager (IRLM).** An MVS subsystem that DB2 uses 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*, *read stability*, *repeatable read*, and *uncommitted read*.

**ISPF.** Interactive System Productivity Facility.

**ISPF/PDF.** Interactive System Productivity Facility/Program Development Facility.

## J

**Japanese Industrial Standards Committee (JISC).** An organization that issues standards for coding character sets.

**JCL.** Job control language.

**JES.** MVS Job Entry Subsystem.

**JIS.** Japanese Industrial Standard.

**job control language (JCL).** A control language that is used to identify a job to an operating system and to describe the job's requirements.

## Job Entry Subsystem (JES) • log

**Job Entry Subsystem (JES).** An IBM licensed program that receives jobs into the system and processes all output data that is produced by the jobs.

**join.** A relational operation that allows retrieval of data from two or more tables based on matching column values. See also *equijoin*, *full outer join*, *inner join*, *left outer join*, *outer join*, and *right outer join*.

## K

**KB.** Kilobyte (1024 bytes).

**Kerberos.** A network authentication protocol that is designed to provide strong authentication for client/server applications by using secret-key cryptography.

**Kerberos ticket.** A transparent application mechanism that transmits the identity of an initiating principal to its target. A simple ticket contains the principal's identity, a session key, a timestamp, and other information, which is sealed using the target's secret key.

**key.** A column or an ordered collection of columns identified in the description of a table, index, or referential constraint.

**key-sequenced data set (KSDS).** A VSAM file or data set whose records are loaded in key sequence and controlled by an index.

**KSDS.** Key-sequenced data set.

## L

**labeled duration.** A number that represents a duration of years, months, days, hours, minutes, seconds, or microseconds.

**large object (LOB).** A sequence of bytes representing bit data, single-byte characters, double-byte characters, or a mixture of single- and double-byte characters. A LOB can be up to 2 GB–1 byte in length. See also *BLOB*, *CLOB*, and *DBCLOB*.

**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 that are being joined, and that 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.

**linkage editor.** A computer program for creating load modules from one or more object modules or load modules by resolving cross references among the modules and, if necessary, adjusting addresses.

**link-edit.** The action of creating a loadable computer program using a linkage editor.

**L-lock.** Logical lock.

**load module.** A program unit that is suitable for loading into main storage for execution. The output of a linkage editor.

**LOB.** Large object.

**LOB table space.** A table space that contains all the data for a particular LOB column in the related base table.

**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, page, or LOB lock to a table space lock because the number of page locks that are 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 that concurrently running programs can have to a resource that a DB2 lock is holding.

**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 LOB, a partition, a table, or a table space.

**log.** A collection of records that describe the events that occur during DB2 execution and that indicate their sequence. The information thus recorded is used for recovery in the event of a failure during DB2 execution.

**logical lock (L-lock).** The lock type that transactions use to control intra- and inter-DB2 data concurrency between transactions. Contrast with *physical lock (P-lock)*.

**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).** The processing that a 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 that DB2 generates and associates with each log record. DB2 also uses the LRSN for page versioning. The LRSNs that a particular DB2 data sharing group generates 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 is to 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. Any LOB column, LOB host variable, or expression that evaluates to a LOB is considered a long string.

**LRECP.** Logical recovery pending.

**LRH.** Log record header.

**LRSN.** Log record sequence number.

**LUW.** Logical unit of work.

**LUWID.** 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.

## logical lock (L-lock) • NUL terminator

**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 that 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 (in IMS).

**MSS.** Mass Storage Subsystem.

**MTO.** Master terminal operator.

**multisite 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

**network identifier (NID).** The network ID that is assigned by IMS or CICS, or if the connection type is RRS/AF, the OS/390 RRS unit of recovery ID (URID).

**NID.** Network ID.

**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'.

## OASN (origin application schedule number) • precompilation

### O

**OASN (origin application schedule number).** In IMS, a 4-byte number that is assigned sequentially to each IMS schedule since the last cold start of IMS. The OASN is used as an identifier for a unit of work. In an 8-byte format, the first 4 bytes contain the schedule number and the last 4 bytes 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.

**OS/390.** Operating System/390.

**OS/390 OpenEdition Distributed Computing Environment (OS/390 OE DCE).** A set of technologies that are provided by the Open Software Foundation to implement distributed computing.

**outer join.** The result of a join operation that includes the matched rows of both tables that are being joined and preserves some or all of the unmatched rows of the tables that are being joined. See also *join*.

### P

**package.** An object containing a set of SQL statements that have been statically bound and that is available for processing. A package is sometimes also called an *application package*.

**package list.** An ordered list of package names that may be used to extend an application plan.

**package name.** The name of an object that is 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 (4 KB, 8 KB, 16 KB, or 32 KB) or index space (4 KB). In a table space, a page contains one or more rows of a table. In a LOB table space, a LOB value can span more than one page, but no more than one LOB value is stored on a page.

**page set.** Another way to refer to a table space or index space. Each page set consists of a collection of VSAM data sets.

**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. The term participant is 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 GB, depending on 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, which are called members. Each partition can contain a program, part of a program, or data. The term partitioned data set is 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 that is subdivided into parts (based on index key range), each of which can be processed independently by utilities.

**partner logical unit.** An access point in the SNA network that is connected to the local DB2 subsystem by way of a VTAM conversation.

**PCT.** Program control table (in 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 for execution.

**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 (in CICS).

**point of consistency.** A time when all recoverable data that an application accesses is consistent with other data. The term point of consistency is synonymous with *sync point* or *commit point*.

**PPT.** (1) Processing program table (in CICS). (2) Program properties table (in 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.** In a relational database, 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.

**principal.** An entity that can communicate securely with another entity. In Kerberos, principals are represented as entries in the Kerberos registry database and include users, servers, computers, and others.

**private connection.** A communications connection that is specific to DB2.

**private protocol access.** A method of accessing distributed data by which you can direct a query to another DB2 system. Contrast with *DRDA access*.

**private protocol connection.** A DB2 private connection of the application process. See also *private connection*.

**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 that are recorded for that ID in the DB2 catalog.

**process.** In DB2, the unit to which DB2 allocates resources and locks. Sometimes called an *application process*, a process involves the execution of one or more programs. The execution of an SQL statement is always associated with some process. The means of initiating and terminating a process are dependent on the environment.

**program.** A single compilable collection of executable statements in a programming language.

**program temporary fix (PTF).** A solution or bypass of a problem that is 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.

**PTF.** Program temporary fix.

## Q

**QMF.** Query Management Facility.

**QSAM.** Queued sequential access method.

**query block.** The part of a query that is represented by one of the FROM clauses. Each FROM clause can have multiple query blocks, depending on DB2's internal processing of the query.

**queued sequential access method (QSAM).** An extended version of the basic sequential access method (BSAM). When this method is used, a queue of data blocks is formed. Input data blocks await processing, and output data blocks await transfer to auxiliary storage or to an output device.

## R

**RACF.** Resource Access Control Facility, which is a component of the SecureWay Security Server for OS/390.

**RBA.** Relative byte address.

**RCT.** Resource control table (in CICS attachment facility).

**RDB.** Relational database.

**RDBMS.** Relational database management system.

**RDBNAM.** 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 that were inserted and committed by a concurrently executing application process.

## rebind • resource control table (RCT)

**rebind.** The creation of a new application plan for an application program that has been bound previously. If, for example, you have added an index for a table that your application accesses, you must rebind the application in order to take advantage of that index.

**rebuild.** The process of reallocating a coupling facility structure. For the shared communications area (SCA) and lock structure, the structure is repopulated; for the group buffer pool, changed pages are usually cast out to DASD, and the new structure is populated only with changed pages that were not successfully cast out.

**record.** The storage representation of a row or other data.

**record identifier (RID) pool.** An area of main storage above the 16-MB 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 indicates their sequence. The recorded information is used for recovery in the event of a failure during DB2 execution.

**recovery pending (RECP).** A condition that prevents SQL access to a table space that needs to be recovered.

**recovery token.** An identifier for an element that is used in recovery (for example, *NID* or *URID*).

**RECP.** Recovery pending.

**redo.** A state of a unit of recovery that indicates that changes 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 state of a database in which all values of all foreign keys are valid. Maintaining referential integrity requires the enforcement of referential constraints on all operations that change the data in a table upon which the referential constraints are defined.

**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 (RDB).** 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 collection of hardware and software that organizes and provides access to a relational database.

**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 that is 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 that is 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 might even operate on the same processor under the same MVS system.

**REORG pending (REORP).** A condition that restricts SQL access and most utility access to an object that must be reorganized.

**REORP.** REORG pending.

**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 that is submitted to the prepare phase if the participant has modified data and is prepared to commit or roll back.

**requester.** The source of a request to access data at a remote server. In the DB2 environment, the requester function is provided by the distributed data facility.

**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 you use to define CICS resources online without assembling tables.

**resource limit facility (RLF).** A portion of DB2 code that prevents dynamic manipulative SQL statements from exceeding specified time limits. The resource limit facility is sometimes called the governor.

**resource limit specification table.** A site-defined table that specifies the limits to be enforced by the resource limit facility.

**restart pending (RESTOP).** A restrictive state of a page set or partition that indicates that restart (backout) work needs to be performed on the object. All access to the page set or partition is denied except for access by the:

- RECOVER POSTPONED command
- Automatic online backout (which DB2 invokes after restart if the system parameter LBACKOUT=AUTO)

**RESTOP.** Restart pending.

**result table.** The set of rows that are 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 that are 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 (in IMS).

**RRSAF.** Recoverable Resource Manager Services attachment facility. RRSAF is 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.

**RS.** Read stability.

**RTT.** Resource translation table.

## 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 that are enclosed in parentheses. Contrast with *column function*.

**SDWA.** System diagnostic work area.

**search condition.** A criterion for selecting rows from a table. A search condition consists of one or more predicates.

**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.** The target of a request from a remote  
| requester. In the DB2 environment, the server function  
| is provided by the distributed data facility, which is used  
| to access DB2 data from remote applications.

**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.

**share lock.** A lock that prevents concurrently executing application processes from changing data, but not from reading data. Contrast with *exclusive lock*.

**short string.** A string whose actual length, or a varying-length string whose maximum length, is 255 bytes (or 127 double-byte characters) or less. Regardless of length, a LOB string is not a short string.

## sign-on • string

**sign-on.** A request that is made on behalf of an individual CICS or IMS application process by an attachment 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 GB, another data set is created, and so on up to a total of 32 data sets. DB2 considers the data sets to be a single contiguous linear address space containing a maximum of 64 GB. 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.  
| Contrast with *double-byte character set* or *multibyte*  
| *character set*.

**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).

**sourced function.** A function that is implemented by another built-in or user-defined function that is already known to the database manager. This function can be a scalar function or a column (aggregating) function; it returns a single value from a set of values (for example, MAX or AVG). Contrast with *built-in function*, *external function*, and *SQL function*.

**source program.** A set of host language statements and SQL statements that is processed by an SQL precompiler.

**special register.** A storage area that DB2 defines for an application process to use for storing information that can be referenced in SQL statements. Examples of special registers are USER and CURRENT DATE.

**SPUFI.** SQL Processor Using File Input.

**SQL.** Structured Query Language.

**SQL authorization ID (SQL ID).** The authorization ID that is used for checking dynamic SQL statements in some situations.

**SQLCA.** SQL communication area.

**SQL communication area (SQLCA).** A structure that is used to provide an application program with information about the execution of its SQL statements.

**SQLDA.** SQL descriptor area.

**SQL descriptor area (SQLDA).** A structure that describes input variables, output variables, or the columns of a result table.

**SQL/DS.** Structured Query Language/Data System. This product is now obsolete and has been replaced by DB2 for VSE & VM.

**SQL function.** A user-defined function in which the CREATE FUNCTION statement contains the source code. The source code is a single SQL expression that evaluates to a single value. The SQL user-defined function can return only one parameter.

**SQL processing conversation.** Any conversation that requires access of DB2 data, either through an application or by dynamic query requests.

**SQL Processor Using File Input (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.

**SRB.** Service request block.

**SSI.** Subsystem interface (in MVS).

**SSM.** Subsystem member.

**stand-alone.** An attribute of a program that means it is capable of executing separately from DB2, without using DB2 services.

**star join.** A method of joining a dimension column of a fact table to the key column of the corresponding dimension table. See also *join*, *dimension*, and *star schema*.

**star schema.** The combination of a fact table (which contains most of the data) and a number of dimension tables. See also *star join*, *dimension*, and *dimension table*.

**statement string.** For a dynamic SQL statement, the character string form of the statement.

**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 that are specified by the statement might change).

**storage group.** A named set of disks on which DB2 data can be stored.

**string.** See *character string* or *graphic string*.

## T

**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 relational database management system (RDBMS).

**sync point.** See *commit point*.

**synonym.** In SQL, an alternative name for a table or view. Synonyms can be used only to refer to objects at the subsystem in which the synonym is defined.

**system administrator.** The person at a computer installation who designs, controls, and manages the use of the computer system.

**system agent.** A work request that DB2 creates internally such as prefetch processing, deferred writes, and service tasks.

**system conversation.** The conversation that two DB2 subsystems 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-directed connection.** A connection that an  
| RDBMS manages by processing SQL statements with  
| three-part names.

**System Modification Program/Extended (SMP/E).** A tool for making software changes in programming systems (such as DB2) 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.

**table.** A named data object consisting of a specific number of columns and some number of unordered rows. See also *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 that is 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 for one of these reasons:

- Each of them contains a table that is a parent or descendent of a table in one of the others.
- The set contains a base table and associated auxiliary tables.

A table space set can contain both types of relationships.

**task control block (TCB).** A control block that is 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 also *address space connection*.

**TB.** Terabyte (1 099 511 627 776 bytes).

**TCB.** Task control block (in MVS).

**temporary table.** A table that holds temporary data; for example, temporary tables are useful for holding or sorting intermediate results from queries that contain a large number of rows. The two kinds of temporary table, which are created by different SQL statements, are the created temporary table and the declared temporary table. Contrast with *result table*. See also *created temporary table* and *declared 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.** 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

## Time-Sharing Option (TSO) • VTAM

unavailability of resources. Installation specifications are set to determine both the amount of time DB2 is to wait for IRLM services after starting, and the amount of time IRLM is to wait if a resource that an application requests is unavailable. If either of these time specifications is exceeded, a timeout is declared.

**Time-Sharing Option (TSO).** An option in MVS that provides interactive time sharing from remote terminals.

**timestamp.** A seven-part value that consists of a date and time. The timestamp is 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

**UDF.** User-defined function.

**undo.** A state of a unit of recovery that indicates that the changes the unit of recovery made to recoverable DB2 resources must be backed out.

**union.** An SQL operation that combines the results of two select statements. Unions are often used to merge lists of values that are obtained from several tables.

**unique constraint.** An SQL rule that no two values in a primary key, or in the key of a unique index, can be the same.

**unique index.** An index which ensures that no identical key values are stored in a table.

**unlock.** The act of releasing an object or system resource that was previously locked and returning 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.

**user-defined function (UDF).** A function that is defined to DB2 by using the CREATE FUNCTION statement and that can be referenced thereafter in SQL statements. A user-defined function can be an *external function*, a *sourced function*, or an *SQL function*. Contrast with *built-in function*.

**UT.** Utility-only access.

## V

**value.** The smallest unit of data that is 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, packages, or LOBs.

**A version of a program** is the source code that is 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 that is 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.

**A version of a LOB** is a copy of a LOB value at a point in time. The version number for a LOB is stored in the auxiliary index entry for the LOB.

**view.** An alternative representation of data from one or more tables. A view can include all or some of the columns that are contained in tables on which it is defined.

**Virtual Storage Access Method (VSAM).** An access method for direct or sequential processing of fixed- and varying-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by relative-record number.

**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.** Virtual Telecommunication Access Method (in MVS).

## W

**WLM application environment.** An MVS Workload Manager attribute that is associated with one or more stored procedures. The WLM application environment determines the address space in which a given DB2 stored procedure runs.

**write to operator (WTO).** An optional user-coded service that allows a message to be written to the system console operator informing the operator of errors and unusual system conditions that may need to be corrected.

**WTO.** Write to operator.

**WTOR.** Write to operator (WTO) with reply.

## X

**XRF.** Extended recovery facility.

## Z

**z/OS.** An operating system for the eServer product line that supports 64-bit real storage.



---

## Bibliography

### **DB2 Universal Database Server for OS/390 and z/OS Version 7 product libraries:**

#### **DB2 for OS/390 and z/OS**

- *DB2 Administration Guide, SC26-9931*
- *DB2 Application Programming and SQL Guide, SC26-9933*
- *DB2 Application Programming Guide and Reference for Java, SC26-9932*
- *DB2 Command Reference, SC26-9934*
- *DB2 Data Sharing: Planning and Administration, SC26-9935*
- *DB2 Data Sharing Quick Reference Card, SX26-3846*
- *DB2 Diagnosis Guide and Reference, LY37-3740*
- *DB2 Diagnostic Quick Reference Card, LY37-3741*
- *DB2 Image, Audio, and Video Extenders Administration and Programming, SC26-9947*
- *DB2 Installation Guide, GC26-9936*
- *DB2 Licensed Program Specifications, GC26-9938*
- *DB2 Master Index, SC26-9939*
- *DB2 Messages and Codes, GC26-9940*
- *DB2 ODBC Guide and Reference, SC26-9941*
- *DB2 Reference for Remote DRDA Requesters and Servers, SC26-9942*
- *DB2 Reference Summary, SX26-3847*
- *DB2 Release Planning Guide, SC26-9943*
- *DB2 SQL Reference, SC26-9944*
- *DB2 Text Extender Administration and Programming, SC26-9948*
- *DB2 Utility Guide and Reference, SC26-9945*
- *DB2 What's New? GC26-9946*
- *DB2 XML Extender for OS/390 and z/OS Administration and Programming, SC27-9949*
- *DB2 Program Directory, GI10-8182*

#### **DB2 Administration Tool**

- *DB2 Administration Tool for OS/390 and z/OS User's Guide, SC26-9847*

#### **DB2 Buffer Pool Tool**

- *DB2 Buffer Pool Tool for OS/390 and z/OS User's Guide and Reference, SC26-9306*

#### **DB2 DataPropagator**

- *DB2 UDB Replication Guide and Reference, SC26-9920*

#### **Net.Data®**

The following books are available at this Web site:  
<http://www.ibm.com/software/net.data/library.html>

- *Net.Data Library: Administration and Programming Guide for OS/390 and z/OS*
- *Net.Data Library: Language Environment Interface Reference*
- *Net.Data Library: Messages and Codes*
- *Net.Data Library: Reference*

#### **DB2 PM for OS/390**

- *DB2 PM for OS/390 Batch User's Guide, SC27-0857*
- *DB2 PM for OS/390 Command Reference, SC27-0855*
- *DB2 PM for OS/390 Data Collector Application Programming Interface Guide, SC27-0861*
- *DB2 PM for OS/390 General Information, GC27-0852*
- *DB2 PM for OS/390 Installation and Customization, SC27-0860*
- *DB2 PM for OS/390 Messages, SC27-0856*
- *DB2 PM for OS/390 Online Monitor User's Guide, SC27-0858*
- *DB2 PM for OS/390 Report Reference Volume 1, SC27-0853*
- *DB2 PM for OS/390 Report Reference Volume 2, SC27-0854*
- *DB2 PM for OS/390 Using the Workstation Online Monitor, SC27-0859*
- *DB2 PM for OS/390 Program Directory, GI10-8223*

#### **Query Management Facility (QMF)**

- *Query Management Facility: Developing QMF Applications, SC26-9579*
- *Query Management Facility: Getting Started with QMF on Windows, SC26-9582*
- *Query Management Facility: High Performance Option User's Guide for OS/390 and z/OS, SC26-9581*
- *Query Management Facility: Installing and Managing QMF on OS/390 and z/OS, GC26-9575*

- *Query Management Facility: Installing and Managing QMF on Windows, GC26-9583*
- *Query Management Facility: Introducing QMF, GC26-9576*
- *Query Management Facility: Messages and Codes, GC26-9580*
- *Query Management Facility: Reference, SC26-9577*
- *Query Management Facility: Using QMF, SC26-9578*

#### **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®**

The following books are available at this Web site:  
[www.as400.ibm.com/infocenter](http://www.as400.ibm.com/infocenter)

- *DB2 Universal Database for AS/400 Database Programming*
- *DB2 Universal Database for AS/400 Performance and Query Optimization*
- *DB2 Universal Database for AS/400 Distributed Data Management*
- *DB2 Universal Database for AS/400 Distributed Data Programming*
- *DB2 Universal Database for AS/400 SQL Programming Concepts*
- *DB2 Universal Database for AS/400 SQL Programming with Host Languages*
- *DB2 Universal Database for AS/400 SQL Reference*

#### **BASIC**

- *IBM BASIC/MVS Language Reference, GC26-4026*
- *IBM BASIC/MVS Programming Guide, SC26-4027*

#### **BookManager READ/MVS**

- *BookManager READ/MVS V1R3: Installation Planning & Customization, SC38-2035*

#### **SAA AD/Cycle C/370**

- *IBM SAA AD/Cycle C/370 Programming Guide, SC09-1841*

- *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 and Registry, SC09-2190*

#### **CICS/ESA**

- *CICS/ESA Application Programming Guide, SC33-1169*
- *CICS External Interfaces Guide, SC33-1944*
- *CICS for MVS/ESA Application Programming Reference, SC33-1170*
- *CICS for MVS/ESA CICS-RACF Security Guide, SC33-1185*
- *CICS for MVS/ESA CICS-Supplied Transactions, SC33-1168*
- *CICS for MVS/ESA Customization Guide, SC33-1165*
- *CICS for MVS/ESA Data Areas, LY33-6083*
- *CICS for MVS/ESA Installation Guide, SC33-1163*
- *CICS for MVS/ESA Intercommunication Guide, SC33-1181*
- *CICS for MVS/ESA Messages and Codes, GC33-1177*
- *CICS for MVS/ESA Operations and Utilities Guide, SC33-1167*
- *CICS/ESA Performance Guide, SC33-1183*
- *CICS/ESA Problem Determination Guide, SC33-1176*
- *CICS for MVS/ESA Resource Definition Guide, SC33-1166*
- *CICS for MVS/ESA System Definition Guide, SC33-1164*
- *CICS for MVS/ESA System Programming Reference, GC33-1171*

#### **CICS Transaction Server for OS/390**

- *CICS Application Programming Guide, SC33-1687*
- *CICS External Interfaces Guide, SC33-1703*
- *CICS DB2 Guide, SC33-1939*
- *CICS Resource Definition Guide, SC33-1684*

#### **IBM C/C++ for MVS/ESA**

- *IBM C/C++ for MVS/ESA Library Reference, SC09-1995*
- *IBM C/C++ for MVS/ESA Programming Guide, SC09-1994*



## **IBM COBOL**

- *IBM COBOL Language Reference, SC26-4769*
- *IBM COBOL for MVS & VM Programming Guide, SC26-4767*

*IBM COBOL for OS/390 & VM Programming Guide, SC26-9049*

## **Conversion Guide**

- *IMS-DB and DB2 Migration and Coexistence Guide, GH21-1083*

## **Cooperative Development Environment**

- *CoOperative Development Environment/370: Debug Tool, SC09-1623*

## **DataPropagator NonRelational**

- *DataPropagator NonRelational MVS/ESA Administration Guide, SH19-5036*
- *DataPropagator NonRelational MVS/ESA Reference, SH19-5039*

## **Data Facility Data Set Services**

- *Data Facility Data Set Services: User's Guide and Reference, SC26-4388*

## **Database Design**

- *DB2 Design and Development Guide by Gabrielle Wiorkowski and David Kull, Addison Wesley, ISBN 0-20158-049-7*
- *Handbook of Relational Database Design by C. Fleming and B. Von Halle, Addison Wesley, ISBN 0-20111-434-8*

## **DataHub®**

- *IBM DataHub General Information, GC26-4874*

## **Data Refresher**

- *Data Refresher Relational Extract Manager for MVS GI10-9927*

## **DB2 Connect**

- *DB2 Connect Enterprise Edition for OS/2 and Windows: Quick Beginnings, GC09-2953*
- *DB2 Connect Enterprise Edition for UNIX: Quick Beginnings, GC09-2952*
- *DB2 Connect Personal Edition Quick Beginnings, GC09-2967*
- *DB2 Connect User's Guide, SC09-2954*

## **DB2 Red Books**

- *DB2 UDB Server for OS/390 Version 6 Technical Update, SG24-6108-00*

## **DB2 Server for VSE & VM**

- *DB2 Server for VM: DBS Utility, SC09-2394*

- *DB2 Server for VSE: DBS Utility, SC09-2395*

## **DB2 Universal Database for UNIX, Windows®, OS/2**

- *DB2 UDB Administration Guide: Planning, SC09-2946*
- *DB2 UDB Administration Guide: Implementation, SC09-2944*
- *DB2 UDB Administration Guide: Performance, SC09-2945*
- *DB2 UDB Administrative API Reference, SC09-2947*
- *DB2 UDB Application Building Guide, SC09-2948*
- *DB2 UDB Application Development Guide, SC09-2949*
- *DB2 UDB CLI Guide and Reference, SC09-2950*
- *DB2 UDB SQL Getting Started, SC09-2973*
- *DB2 UDB SQL Reference Volume 1, SC09-2974*
- *DB2 UDB SQL Reference Volume 2, SC09-2975*

## **Device Support Facilities**

- *Device Support Facilities User's Guide and Reference, GC35-0033*

## **DFSMS**

These books provide information about a variety of components of DFSMS, including DFSMS/MVS, DFSMSdftp, DFSMSdss™, DFSMSShsm, and MVS/DFP™.

- *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: DFSMSShsm Managing Your Own Data, SH21-1077*
- *DFSMS/MVS: Diagnosis Reference for DFSMSdftp, LY27-9606*
- *DFSMS/MVS Storage Management Library: Implementing System-Managed Storage, SC26-3123*
- *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 DFSMSdftp, SC26-4920*
- *DFSMS/MVS: Using Advanced Services, SC26-4921*

- *DFSMS/MVS: Utilities*, SC26-4926
- *MVS/DFP: Using Data Sets*, SC26-4749

## **DFSORT**

- *DFSORT Application Programming: Guide*, SC33-4035

## **Distributed Relational Database Architecture™**

- *Data Stream and OPA Reference*, SC31-6806
- *IBM SQL Reference*, SC26-8416
- *Open Group Technical Standard*

The Open Group presently makes the following DRDA books available through its Web site at: [www.opengroup.org](http://www.opengroup.org)

- *DRDA Version 2 Vol. 1: Distributed Relational Database Architecture (DRDA)*
- *DRDA Version 2 Vol. 2: Formatted Data Object Content Architecture*
- *DRDA Version 2 Vol. 3: Distributed Data Management Architecture*

## **Domain Name System**

- *DNS and BIND, Third Edition*, Paul Albitz and Cricket Liu, O'Reilly, ISBN 1-56592-512-2

## **Education**

- *IBM Dictionary of Computing*, McGraw-Hill, ISBN 0-07031-489-6
- *1999 IBM All-in-One Education and Training Catalog*, GR23-8105

## **Enterprise System/9000® and Enterprise System/3090™**

- *Enterprise System/9000 and Enterprise System/3090 Processor Resource/System Manager Planning Guide*, GA22-7123

## **High Level Assembler**

- *High Level Assembler for MVS and VM and VSE Language Reference*, SC26-4940
- *High Level Assembler for MVS and VM and VSE Programmer's Guide*, SC26-4941

## **Parallel Sysplex Library**

- *OS/390 Parallel Sysplex Application Migration*, GC28-1863
- *System/390 MVS Sysplex Hardware and Software Migration*, GC28-1862
- *OS/390 Parallel Sysplex Overview: An Introduction to Data Sharing and Parallelism*, GC28-1860
- *OS/390 Parallel Sysplex Systems Management*, GC28-1861
- *OS/390 Parallel Sysplex Test Report*, GC28-1963

- *System/390 9672/9674 System Overview*, GA22-7148

## **ICSF/MVS**

- *ICSF/MVS General Information*, GC23-0093

## **IMS**

- *IMS Batch Terminal Simulator General Information*, GH20-5522
- *IMS Administration Guide: System*, SC26-9420
- *IMS Administration Guide: Transaction Manager*, SC26-9421
- *IMS Application Programming: Database Manager*, SC26-9422
- *IMS Application Programming: Design Guide*, SC26-9423
- *IMS Application Programming: Transaction Manager*, SC26-9425
- *IMS Command Reference*, SC26-9436
- *IMS Customization Guide*, SC26-9427
- *IMS Install Volume 1: Installation and Verification*, GC26-9429
- *IMS Install Volume 2: System Definition and Tailoring*, GC26-9430
- *IMS Messages and Codes*, GC27-1120
- *IMS Utilities Reference: System*, SC26-9441

## **ISPF**

- *ISPF V4 Dialog Developer's Guide and Reference*, SC34-4486
- *ISPF V4 Messages and Codes*, SC34-4450
- *ISPF V4 Planning and Customizing*, SC34-4443
- *ISPF V4 User's Guide*, SC34-4484

## **Language Environment**

- *Debug Tool User's Guide and Reference*, SC09-2137

## **National Language Support**

- *IBM National Language Support Reference Manual Volume 2*, SE09-8002

## **NetView**

- *NetView Installation and Administration Guide*, SC31-8043
- *NetView User's Guide*, SC31-8056

## **Microsoft® ODBC**

- *Microsoft ODBC 3.0 Software Development Kit and Programmer's Reference*, Microsoft Press, ISBN 1-57231-516-4

## **OS/390**

- *OS/390 C/C++ Programming Guide*, SC09-2362
- *OS/390 C/C++ Run-Time Library Reference*, SC28-1663

- *OS/390 C/C++ User's Guide, SC09-2361*
- *OS/390 eNetwork Communications Server: IP Configuration, SC31-8513*
- *OS/390 Hardware Configuration Definition Planning, GC28-1750*
- *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 MVS Using the Subsystem Interface, SC28-1789*
- *OS/390 Security Server External Security Interface (RACROUTE) Macro Reference, GC28-1922*
- *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) Introduction, GC28-1912*
- *OS/390 Security Server (RACF) Macros and Interfaces, SK2T-6700 (OS/390 Collection Kit ), SK27-2180 (OS/390 Security Server Information Package )*
- *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 Support for Unicode: Using Conversion Services, SC33-7050*
- *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 DCE Administration Guide, SC28-1584*
- *OS/390 DCE Introduction, GC28-1581*
- *OS/390 DCE Messages and Codes, SC28-1591*
- *OS/390 UNIX System Services Command Reference, SC28-1892*
- *OS/390 UNIX System Services Messages and Codes, SC28-1908*
- *OS/390 UNIX System Services Planning, SC28-1890*
- *OS/390 UNIX System Services User's Guide, SC28-1891*
- *OS/390 UNIX System Services Programming: Assembler Callable Services Reference, SC28-1899*

#### **IBM Enterprise PL/I for z/OS and OS/390**

- *IBM Enterprise PL/I for z/OS and OS/390 Language Reference, SC26-9476*
- *IBM Enterprise PL/I for z/OS and OS/390 Programming Guide, SC26-9473*

#### **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*

#### **RAMAC and Enterprise Storage Server™**

- *IBM RAMAC Virtual Array, SG24-4951*
- *RAMAC Virtual Array: Implementing Peer-to-Peer Remote Copy, SG24-5338*
- *Enterprise Storage Server Introduction and Planning, GC26-7294*

#### **Remote Recovery Data Facility**

- *Remote Recovery Data Facility Program Description and Operations, LY37-3710*

#### **Storage Management**

- *DFSMS/MVS Storage Management Library: Implementing System-Managed Storage, SC26-3123*
- *MVS/ESA Storage Management Library: Leading a 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**

- *ESA/370 Principles of Operation, SA22-7200*
- *ESA/390 Principles of Operation, SA22-7201*
- *System/390 MVS Sysplex Hardware and Software Migration, GC28-1210*

#### **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*

#### **VS COBOL II**

- *VS COBOL II Application Programming Guide for MVS and CMS, SC26-4045*
- *VS COBOL II Application Programming: Language Reference, GC26-4047*
- *VS COBOL II Installation and Customization for MVS, SC26-4048*

#### **VS Fortran**

- *VS Fortran Version 2: Language and Library Reference, SC26-4221*
- *VS Fortran Version 2: Programming Guide for CMS and MVS, SC26-4222*

#### **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

32-KB buffering 255

4-KB buffering 255

## A

ACBNAME option of APPL statement 477

ACCEPT job 54, 66

active log

data set

disk requirements 27

prefix 105

sharing DASD 266

dual logging 103

installation 240

size

estimating 27, 130

address space

allied agent 37

database services

description 37

working storage calculation 48

DDF 36

DSN1DIST 36

IRLM 36

system services 37

ADSNBASE library 52

ADSNDKF library 52

ADSNENU library 52

ADSNIVPD library 52

ADSNLOAD library 52

ADSNLOD2 library 52

ADSNMACS library 52

ADXRLOAD library 52

ADXRSAMP library 52

AIHSCLSB library 52

AIHSMODB library 52

AIHSSMPB library 52

alias

migration considerations 282

allied agent address space 37

allocation job 62, 64

ALLOCATION UNITS field of panel DSNTIPA 197

AMDPRECT module 236

APN option of DSNMAPN macro 430, 431

APPC option of APPL statement 476

APPL REGISTRATION TABLE field of panel

DSNTIPZ 218

APPL statement 473

example 473

option

description 474

options

DSESLIM 496

LUNAME 507

APPLICATION ENCODING field of panel

DSNTIPF 166

APPLICATION LOAD field of panel DSNTIPT 109

APPLY job 65

archive log

cataloging options 97

data set

disk requirements 35

prefix 105

sharing DASD 266

dual logging 103

ARCHIVE LOG FREQ field of panel DSNTIPL 193

ARCHIVE LOG RACF field of panel DSNTIPP 184

ART/ORT ESCAPE CHARACTER field of panel

DSNTIPZ 217

ASCII

conversion table 524

ASCII CODED CHARACTER SET field of panel

DSNTIPF 165

ASSEMBLER field of panel DSNTIPG 123

ASSISTANT field of panel DSNTIPK 102

ATCSTRxx member of SYS1.VTAMLST library 485

ATNLOSS option of APPL statement 476

attachment facility

CICS 253, 292, 329

IMS

installation 253, 425

migration 292, 329

TSO 246, 289, 327

AUDIT TRACE field of panel DSNTIPN 149

AUTH AT HOP SITE field of panel DSNTIP5 212

AUTH MEMBER field of panel DSNTIPM 190

AUTH option

APPL statement 476

DSNCRCT macro

TYPE=ENTRY 454

AUTH SEQUENCE field of panel DSNTIPM 190

AUTHID

column of MODESELECT table 491

column of USER NAMES catalog table 518

authorization ID

RACF 253

TSO 253

AUTO BIND

field of panel DSNTIPO 312, 347

AUTO BIND field of panel DSNTIPO 155

AUTO START field of panel DSNTIPI 175

automatic

recall 153

AUTOSSES option of APPL statement 474

## B

BACKOUT DURATION field of panel DSNTIPL 195

BIND NEW PACKAGE field of panel DSNTIPP 186

binding

installation 257

migration

DCLGEN 301, 339

SPUFI 301, 339

- binding (*continued*)
  - remote package
    - plan name for 491
    - relation to character conversion 531
- BLOCK SIZE field of panel DSNTIPA 199
- bootstrap data set (BSDS) 294, 332
- BP0—BP25 fields of panel DSNTIP1 144
- BP16K0—BP16K9 fields of panel DSNTIP6 147
- BP26—BP49 fields of panel DSNTIP2 145
- BP32K—BP32K9 fields of panel DSNTIP6 147
- BP8K0—BP8K9 fields of panel DSNTIP6 147
- BSDS (bootstrap data set)
  - adding a second 294, 332
  - installation 240
  - space requirement 29
- buffer pool
  - storage requirement 40
  - VTAM IOBUF storage requirements 495

## C

- C++ fields of panel DSNTIPU 119
- C fields of panel DSNTIPU 115, 116
- CACHE DYNAMIC SQL field of panel DSNTIP4 170
- capturing changed data
  - log data set size 29
- catalog, DB2
  - disk requirements 26
  - installing 240
- catalog, integrated catalog facility 240
- CATALOG ALIAS field of panel DSNTIPA2 96
- CATALOG DATA field of panel DSNTIPA 198
- catalog tables
  - SYS PROCEDURES
    - migration considerations 276
  - SYSSTRINGS
    - description 529
- CATMAINT utility
  - migration from Version 5 300
  - migration from Version 6 337
- CCSID (coded character set identifier)
  - code page 527
  - definition 524
  - installation option for 164
  - specifying 524
- CD-ROM, books on xvii
- CDB (communications database)
  - creating
    - during installation 258
  - creating during installation 257
  - description 480, 516
  - dropping while DDF is active 491
  - IPNAMES table 465, 517
  - LOCATIONS table 465, 480, 484, 516
  - LULIST table 480, 484
  - LUMODES table 484
  - LUNAMES table 465, 481, 484
  - MODESELECT table 484
  - populating while
    - connecting DB2 subsystems 480, 516
    - installing 259

- CDB (communications database) (*continued*)
  - updating while DDF is active 491
  - USERNAMES table 465, 483, 484, 518
- change log inventory utility
  - changing
    - generic LU name 507
    - location name 507
    - LU name 507
    - VTAM password 507
- changed data capture 29
- channel-to-channel (CTC) 502
- character conversion
  - code page 527
  - code point 527
  - contracting conversion 527
  - description 523
  - distributed data 523
  - expanding conversion 527
  - locale
    - definition 531
    - specifying 531
    - support for euro currency 531
  - rebinding remote packages 531
  - SYSIBM.SYSSTRINGS catalog table 529
  - Unicode 527
- character string
  - transmitting 523
- checkpoint
  - frequency 29
- CHECKPOINT FREQ field of panel DSNTIPL 193
- CICS
  - attachment facility
    - connecting 433
  - commands
    - DSNCCOM1 transaction program 441
  - connecting to DB2
    - accessing resources 443
    - authorization 442
    - authorization IDs 448
    - installation 253
    - program definitions 433
    - security 442
  - facilities
    - message destination 446
    - resource control table 436
    - subtask failure snap dump 448
  - initialization JCL 442
  - migration 292, 329, 433
  - operating
    - create thread error processing 446
    - starting 448
    - starting an application 380
    - testing 377
    - using 380
  - planning
    - OSCOR storage area 436
    - storage requirements 433
  - system tables
    - program entries 439
    - program list table 440
    - system initialization table 436

- CICS (*continued*)
  - system tables (*continued*)
    - transaction entries 437
  - table generation procedures 444
  - thread
    - maximum, specifying 449, 451
- CICS COBOL II LIBRARY field of panel DSNTIPW 129
- CICS COBOL LIBRARY field of panel DSNTIPW 129
- CICS EXCI LIBRARY field of panel DSNTIPW 129
- CICS LOAD LIBRARY field of panel DSNTIPW 129
- CICS MACRO LIBRARY field of panel DSNTIPW 129
- CICS PL/I LIBRARY field of panel DSNTIPW 129
- class of service, modifying 488
- CLIST LIBRARY field of panel DSNTIPT 109
- CLIST messages fields of panel DSNTIPC1 227
- CLUSTERRATIOF column
  - SYSINDEXES catalog table
    - migration consideration 283
  - SYSINDEXSTATS catalog table
    - migration consideration 283
- CNOS (change number of sessions)
  - description 496
- COBOL TYPE field of panel DSNTIPY 220
- code page 527
- coded character set 523
- CODED CHARACTER SET fields of panel DSNTIPF 164
- COLUMNS field of panel DSNTIPD 131
- command recognition character (CRC) 427
- common service area (CSA) 35
- communications database (CDB) 465, 481
- COMPACT DATA field of panel DSNTIPA 202
- compression
  - archive log data set 202
- COMPROT option of MODEENT macro 480
- connection
  - database management systems 465, 481, 517
  - DB2 481, 517
    - systems 465
  - DB2I panels to ISPF
    - installation 249
    - migration 291, 328
- connection exit routine
  - installation 244
  - migration 297, 335
- contention
  - for conversation 474
- CONTRACT THREAD STG field of panel DSNTIPE 142
- CONTROL ALL APPLICATIONS field of panel DSNTIPZ 216
- Control Center, DB2 UDB 268
- conversation
  - contention 474
  - definition 469
  - SQL 489
  - system 482
- conversion procedure
  - conversion table 523

- CONVLIMIT column of LUMODES catalog table
  - CNOS negotiation 496
  - description 490
- COORDINATOR field of panel DSNTIPK 102
- COPY 1 NAME field of panel DSNTIPH 104
- COPY 1 PREFIX field of panel DSNTIPH 104, 105
- COPY 2 NAME field of panel DSNTIPH 104
- COPY 2 PREFIX field of panel DSNTIPH 105
- CRC (command recognition character)
  - installation 427
- CROSS MEMORY
  - IRLM 36
- CROSS MEMORY field of panel DSNTIPJ 179
- CSA (common service area) 35, 37
- CTC (channel-to-channel)
  - considerations for MAXBFRU size 502
  - sample definitions for VTAM 502
  - X' sense code 502

## D

- data
  - compression 421
  - distributed 465
- data compression
  - Huffman 421
- Data Facility Sort (DFSORT) 238
- data set
  - control block size calculation 47
  - storage requirements 26
- DATA SET NAME(MEMBER) field of panel DSNTIPA1 92
- DATA SHARING field of DSNTIPA1 91
- database
  - DSNDB06 (DB2 catalog database) 240
  - DSNDB07 (work file database) 255
  - temporary storage estimation 30
- DATABASE PROTOCOL field of panel DSNTIP5 211
- database services address space 37
- DATABASES field of panel DSNTIPD 131
- DATABASES field of panel DSNTIPE 138
- DATASET STATS TIME field of panel DSNTIPN 151
- DATE FORMAT field of panel DSNTIP4 168
- DATE
  - option of DSNHDECP macro 168
- DB2 books online xvii
- DB2 Connect
  - creating database during installation 259
  - storage estimation 35
- DB2 GENERIC LUNAME
  - field of panel DSNTIPR 472
- DB2 GENERIC LUNAME field of panel DSNTIPR 208
- DB2 Installer 7
- DB2 LOCATION NAME field of panel DSNTIPR 206
- DB2 NETWORK LUNAME field of panel DSNTIPR 206
- DB2 NETWORK PASSWORD field of panel DSNTIPR 206
- DB2 Online Help
  - accessing 75
  - copying the bookshelf list 71

DB2 Online Help (*continued*)

- customizing BookManager READ 72
- exiting 76
- installation 71
- reading 75
- searching text 75
- using 74

DB2 private protocol access

- definition 465
- setting up 465

DB2 PROC NAME field of panel DSNTIPX 214

DB2 UDB Control Center 268

DB2I (DB2 Interactive)

- connecting to ISPF
- installation 249
- migration 291, 328
- English panels, fall back 70
- Kanji panels, fall back 70

DBADM CREATE AUTH field of panel DSNTIPP 187

DBCS (double-byte character set)

- identifiers 525

DBD (database descriptor)

- size 46

DBPROTOCOL

- migration consideration 283

DBRM LIBRARY field of panel DSNTIPT 109

DBRMLIB.DATA library data set

- DASD volume 99
- device type 99
- DSNTIJIN job 240
- installing a second DB2 263, 265
- naming considerations 61

DCLGEN (declarations generator)

- installation 370
- migration 301, 339

DDCS (data definition control support)

- creating database during installation 259
- storage estimation 35

DDF (distributed data facility)

- address space 36
- overview 465

DDF STARTUP OPTION field of panel DSNTIPR 205

DDF THREADS field of panel DSNTIPR 207

DDRAINL option of APPL statement 477

deadlock

- cycles 181
- sync point rollback 447

DEADLOCK CYCLE field of panel DSNTIPJ 181

DEADLOCK TIME field of panel DSNTIPJ 181

DEALLOC PERIOD field of panel DSNTIPA 200

DECIMAL ARITHMETIC field of panel DSNTIPF 166

DECIMAL POINT IS field of panel DSNTIPF 161

DECLARATION LIBRARY field of panel DSNTIPT 109

declared temporary table

- for scrollable cursor 32

DEF ENCODING SCHEME field of panel DSNTIPF 165

DEFAULT BUFFER POOL FOR USER DATA field of panel DSNTIP1 143

DEFAULT BUFFER POOL FOR USER INDEXES field of panel DSNTIP1 143

default database (DSNDB04)

- storage estimation 32

DEFER field of panel DSNTIPA 203

DEFINE CATALOG field of panel DSNTIPA2 98

deleting

- data sets, DSNTIJDE 241

DESCRIBE FOR STATIC field of panel DSNTIPF 167

DEVICE TYPE 1 field of panel DSNTIPA 198

DEVICE TYPE 2 field of panel DSNTIPA 199

DFSESL DD statement 59, 425

DFSMS (Data Facility Storage Management Subsystem)

- installation 184

DFSMSshm (Data Facility Hierarchical Storage Manager)

- RECALL command 153

DFSORT (Data Facility Sort)

- program library 238

directory

- disk requirements 26
- installing 240
- panel field names 82
- panels 81

DISCONNECT IRLM field of panel DSNTIPJ 183

DISPLAY NET,BFRUSE command of VTAM 485

DIST SQL STR DELIMTR field of panel DSNTIPF 163

distributed data

- planning
  - DB2 private protocol access 465
  - DRDA access 465
  - number of systems that can be connected 465
- programming
  - character conversion 523
- testing 383

distributed unit of work 465

distribution libraries

- manage using DFSMSshm 153
- SMP/E 52

DL/I BATCH TIMEOUT field of panel DSNTIPI 178

DMINWNL option of APPL statement 474

DMINWNR option of APPL statement 474

domain name definition 509

domain name server

- definition 509

DPMODE option of DSNCRCCT macro 456

DPMODI option of DSNCRCCT macro 445

DPROP SUPPORT field of panel DSNTIPO 157

DRDA (Distributed Relational Database Architecture)

- remote access control 465

DRDA access

- definition 465
- organization application 409
- setting up 465
- specifying modes 489
- updating 411

DRDA PORT field of panel DSNTIP5 210

DRDA RDBNAM (relational database name)

- description 471

DRESPL option of APPL statement 477

DSESLIM option of APPL statement

- CNOS negotiation 496
- description 474



DSMAX field of panel DSNTIPC 223  
 DSMAX limit on open data sets, description 47  
 DSN1CHKR utility  
     migration preparation 286, 324  
 DSN1COPY utility  
     migration preparation 286, 324  
 DSN1DIST address space 36  
 DSN3@ATH connection exit routine 244  
 DSN3@SGN sign-on exit routine 244  
 DSN3EPX load module 237  
 DSN3INI load module 237  
 DSN6ARVP macro 242  
 DSN6ENV macro 242  
 DSN6FAC macro 242  
 DSN6GRP macro 242  
 DSN6LOGP macro 242  
 DSN6SPRM macro  
     installation 242  
 DSN6SYSP  
     macro 242  
 DSN8EAE1 exit routine 420  
 DSN8HUFF edit routine 421  
 DSNACEP1 job 54, 66  
 DSNACEP2 job 54, 66  
 DSNACICS stored procedure 257  
 DSNACICX exit routine 244  
 DSNALLOC job 54, 64  
 DSNAPPL1 job 54, 65  
 DSNAPPL2 job 54, 65  
 DSNBIND plan name in SYSIBM.MODESELECT  
     table 491  
 DSNCONNS ISPF table 248  
 DSNCRCT macro  
     description 444  
     specification requirements 444  
     TYPE=COMD 451  
     TYPE=DSECT 461  
     TYPE=ENTRY  
         description 453  
         null specification (\*) 455  
         option descriptions 454, 460  
     TYPE=FINAL 461  
     TYPE=INIT  
         description 444  
         option descriptions 445, 450  
         SUFFIX option 442  
     TYPE=POOL  
         description 452  
         option descriptions 452, 453  
 DSNDB06 database  
     installation job DSNTIJIN 240  
 DSNDBRMS ISPF table 248  
 DSNDDF database 465  
 DSNEMCO1 CLIST 246  
 DSNPRI panel of ISPF 252  
 DSNBTBLS data set for ISPF tables 248  
 DSNH migration considerations 283  
 DSNHASP procedure 239  
 DSNHC procedure 239  
 DSNHCOB procedure 239  
 DSNHCOB2 procedure 239  
 DSNHCPP procedure 239  
 DSNHCPP2 procedure 239  
 DSNHDECP  
     installation 242  
     list of parameters 533  
     migration 293, 331  
 DSNHFOR procedure 239  
 DSNHICB2 procedure 239  
 DSNHICOB procedure 239  
 DSNHPLI procedure 239  
 DSNHSQL procedure 239  
 DSNMAPN macro 430, 431  
 DSNPLPKN ISPF table 248  
 DSNRECV1 job 54, 64  
 DSNRECV2 job 54, 64  
 DSNRECV3 job 54, 64  
 DSNRECV4 job 54, 64  
 DSNTJxx job  
     installation 350  
     migration 11, 303, 340  
 DSNTESA job 370  
 DSNTESC job 370  
 DSNTESJ job 370  
 DSNTESQ queries 288, 325, 423  
 DSNTIAC subroutine  
     calling 443  
 DSNTIAD sample program  
     executes SQL statements 423  
     invoked by DSNTJ1 360  
     run by DSNTIJTM 255, 301, 338  
 DSNTIDxx member 95  
 DSNTIJAA job 54, 62  
 DSNTIJAL job 62, 64  
 DSNTIJCA job 240  
 DSNTIJCC job 268  
 DSNTIJDE job 241  
 DSNTIJEX job  
     installation 244  
     migration 297, 335  
 DSNTIJFV job 310, 345  
 DSNTIJIC job  
     installation 259  
     migration 288, 302, 326, 340  
 DSNTIJID job 241  
 DSNTIJIN job 240, 297, 335  
 DSNTIJMV job 235, 295, 332  
 DSNTIJPM job 279  
 DSNTIJSE job 54  
 DSNTIJSG job  
     installation 257  
     migration 301, 339  
 DSNTIJSG sample program 301  
 DSNTIJTC job 300, 337  
 DSNTIJTM job 255, 301, 338  
 DSNTIJUD job 54, 65  
 DSNTIJUZ job 242, 293, 331  
 DSNTIJVC job 246, 290, 327  
 DSNTINS0 CLIST 77, 78  
 DSNTINST CLIST 7, 78  
 DSNTIP1 installation panel 143  
 DSNTIP2 installation panel 145

- DSNTIP4 installation panel 168
- DSNTIP5 installation panel 210
- DSNTIP6 installation panel 147
- DSNTIP7 installation panel 136
- DSNTIPA installation panel 197
- DSNTIPA0 installation panel 89
- DSNTIPA1 installation panel 90
- DSNTIPA2 installation panel 96
- DSNTIPB installation panel 231, 232
- DSNTIPC installation panel 223
- DSNTIPC1 installation panel 227
- DSNTIPD installation panel 131
- DSNTIPE installation panel 138
- DSNTIPF installation panel 161
- DSNTIPG installation panel 123
- DSNTIPH installation panel 104
- DSNTIPI installation panel 174
- DSNTIPJ installation panel 179
- DSNTIPK installation panel 101
- DSNTIPL installation panel 192
- DSNTIPM installation panel 188
- DSNTIPN installation panel 149
- DSNTIPO installation panel 153
- DSNTIPP installation panel 184
- DSNTIPQ installation panel 120
- DSNTIPR installation panel 205
- DSNTIPS installation panel 203
- DSNTIPT installation panel 108
- DSNTIPU installation panel 115
- DSNTIPW installation panel 127
- DSNTIPX installation panel 214
- DSNTIPY installation panel 220
- DSNTIPZ installation panel 216
- DSNTNJxx jobs 54, 55
- DSNTPSMP stored procedure
  - description 257
- DSNUTILB entry in PPT 235
- DSNUTILS stored procedure
  - description 257
  - sample application 387
- DSNWZP stored procedure 257
- DSNYASCP entry in PPT 235
- DSNZPARM
  - list of parameters 533
- dual logging
  - specifying 103
- dump
  - data set size 34
- DXRRLM00 entry in PPT 235
- dynamic plan selection in CICS
  - DSNCRCT macro 456
- dynamic SQL
  - DSNTESA 421
  - DSNTESQ 421
  - programs using 423
- DYNAMICRULES
  - migration considerations 283

## E

- early code 57
- EAS option of APPL statement 477
- ECSA (extended common storage area) 180
- edit routine
  - DSN8HUFF 421
- EDM pool
  - package size 43
  - plan size 43
  - size
    - calculation 43
- EDMPOOL DATA SPACE SIZE field of panel
  - DSNTIPC 224
- EDMPOOL STORAGE SIZE field of panel
  - DSNTIPC 223
- ENCR option of APPL statement 476
- ENCRYPTPSWDS column of LUNAMES catalog
  - table 482
- END
  - option of DSNMAPN macro
    - description 432
    - installation format 430
- ERLY code 57
- ERRDEST option
  - DSNCRCT macro 446
- euro currency support 531
- EVALUATE UNCOMMITTED field of panel
  - DSNTIP4 173
- EXECUTED STMTS field of panel DSNTIPD 133
- EXIT LIBRARY field of panel DSNTIPT 110
- exit routine
  - description 420
- EXPLAIN PROCESSING field of panel DSNTIPO 157
- extended common storage area (ECSA) 180
- extended English code page 527
- extended Katakana code page 527
- EXTENDED SECURITY field of panel DSNTIPR 209
- external storage 25
- EXTRA BLOCKS REQ field of panel DSNTIP5 211
- EXTRA BLOCKS SRV field of panel DSNTIP5 211

## F

- fallback
  - automatic rebind 306, 343
  - description 308, 344
  - frozen objects 304, 342
  - jobs 310, 345
  - release incompatibilities 306, 343
  - remigration following 312, 347
  - steps 12
- FMIDs 51
- FMPROF option of MODEENT macro 480
- FORMAT
  - command in sample application 376, 377
- FORTTRAN COMPILER field of panel DSNTIPG 124
- FORTTRAN LINK EDIT field of panel DSNTIPG 124
- FREQUENCY TYPE field of panel DSNTIPL 194
- FROM RELEASE field of panel DSNTIPA1 92
- frozen objects 304, 342

function keys 398

## G

GDDM LOAD MODULES field of panel DSNTIPW 128  
GDDM MACLIB field of panel DSNTIPW 128  
GENERIC column of LUNAMES catalog table  
    description 483  
    installation panel 472  
global deadlock cycle 181  
global trace 149  
governor (resource limit facility) 154  
graphic coded character set identifiers 526  
GROUP ATTACH field of panel DSNTIPK 102  
GROUP NAME field of panel DSNTIPK 101  
GROUP option  
    DSNCRCT macro 455

## H

HAVAIL option of APPL statement 476  
help  
    online 71  
Hierarchical Storage Manager (DFSMSHsm) 153  
hiperpool  
    storage requirement 41  
Huffman compression  
    exit routine 421

## I

IBM COBOL fields of DSNTIPQ 121, 122  
IBMDB2LM mode 478  
IBMRDB mode 478  
IBMUSER authority, establishing authorization ID 253  
identity column  
    migration considerations 322  
IDLE THREAD TIMEOUT  
    field of panel DSNTIPR 519  
IDLE THREAD TIMEOUT field of panel DSNTIPR 208  
IDRC 202  
IEAAPFxx member of SYS1.PARMLIB  
    DSNTIJMV job  
        installation 237  
        migration 296, 333  
    DSNTIPM panel 188  
IEBCOPY utility 55  
IEBPTPCH utility 55  
IEFSSNxx member of SYS1.PARMLIB  
    DSNTIJMV job  
        installation 236  
        migration 295, 333  
    DSNTIPM panel 188  
IMMEDIATE WRITE field of panel DSNTIP4 172  
IMS  
    attachment facility 425  
    connecting to DB2  
        installation 253, 425  
    language interface module (DFSLI000)  
        generating 430  
    migration 292, 329

IMS (*continued*)

    operating  
        starting 376  
        testing 373

IMS BMP TIMEOUT field of panel DSNTIPI 178

IMS.PROCLIB library  
    description 426

IMS RESLIB field of panel DSNTIPW 128

IMSCRTL macro 426

incompatibilities of releases 281, 321

initial program load (IPL) 253

INPUT MEMBER NAME field of panel DSNTIPA1 94

INSTALL DD CONTROL SUPT. field of panel  
    DSNTIPZ 216

INSTALL IRLM field of panel DSNTIPI 174

INSTALL TYPE field of panel DSNTIPA1 90

installation

    CLIST 7, 77

    defining DB2 to MVS 235

    description 8

    IRLM 174, 179

    jobs

        description 7

        DSNTIJCA 240

        DSNTIJDE 241

        DSNTIJEX 244

        DSNTIJIC 259

        DSNTIJID 241

        DSNTIJIN 240

        DSNTIJMV 235, 295, 332

        DSNTIJSG 257

        DSNTIJTM 255, 301, 338

        DSNTIJUZ 242

        DSNTIJVC 246

        system affinity 219

    macros

        DSN6ARVP 242

        DSN6ENV 242

        DSN6FAC 242

        DSN6GRP 242

        DSN6LOGP 242

        DSN6SPRM 242

        DSN6SYSP 242

    output 78

    overview 261

    panels

        description 80

        list of 81

    planning 7

    second subsystem 261

    steps 10

    tailoring session 8

    tapes or cartridges 51

    verification

        planning 354

        testing batch environment 364

        testing CICS environment 377

        testing IMS environment 373

        testing PL/I batch 369

        testing SPUFI 369

        testing the TSO attachment facility 360

- installation CLIST
  - calculating disk requirements 35
- installation SYSADM authority
  - authorization IDs established 253
- installation SYSOPR authority
  - authorization IDs established 253
- installation tapes or cartridges 51
- installation verification procedure (IVP) 312, 347
- integrated catalog facility
  - DSNTIJCA job 240
- invoking
  - CLIST 78
- IOBUF buffer pool
  - calculating storage requirements 495
  - description 485
- IP address, definition 509
- IPADDR column of IPNAMES catalog table 517
- IPL (initial program load)
  - installation 253
  - migration 298, 336
- IRLM
  - address space 239
  - fallback, stopping IRLM during 309, 344
  - FMID 51
  - group naming, naming convention 101
  - installation
    - AUTO START parameter 174
    - CROSS MEMORY parameter 179
    - DEADLOCK CYCLE parameter 179
    - DEADLOCK TIME parameter 179
    - DISCONNECT IRLM parameter 179
    - DL/I BATCH TIMEOUT parameter 174
    - IMS BMP TIMEOUT parameter 174
    - INSTALL IRLM parameter 174
    - IRLM LOCK MAXIMUM SPACE parameter 222
    - IRLM XCF GROUP NAME parameter 179
    - LOCK ENTRY SIZE parameter 179
    - LOCKS PER TABLE(SPACE) parameter 179
    - LOCKS PER USER parameter 179
    - MAXIMUM ECSA parameter 179
    - MEMBER IDENTIFIER parameter 179
    - panel DSNTIPI 174
    - panel DSNTIPJ 179
    - PROC NAME parameter 174
    - RESOURCE TIMEOUT parameter 174
    - START IRLM CTRACE parameter 174
    - SUBSYSTEM NAME parameter 174
    - TIME TO AUTOSTART parameter 174
    - U LOCK FOR RR/RS parameter 174
    - UTILITY TIMEOUT parameter 174
  - installing a second DB2 subsystem 261
  - libraries
    - load 107
    - sample 52
    - target 53
  - loading DB2 libraries
    - maintenance considerations 65
    - migration considerations 65
  - MVS system linkage index 235
  - naming convention for group names 101
  - space, estimating maximum 222

- IRLM (*continued*)
  - SYS1.PARMLIB updates 236
- IRLM (internal resource lock manager)
  - address space (IRLMPROC) 36
  - dump formatting module, AMDPRECT 236
  - entry in PPT 235
  - starting
    - after fallback 310, 346
    - installation 254
    - migration 299, 336
- IRLM LOCK MAXIMUM SPACE field of panel DSNTIPC 222
- IRLM XCF GROUP NAME field of panel DSNTIPJ 182
- IRLMPROC 36, 239
- ISPF (Interactive System Productivity Facility)
  - primary option menu, panel connection 249
- ISPF ISPLINK MODULE field of panel DSNTIPW 128
- ISPF Skeleton Library (ISPSLIB) 77
- ISTINCLM mode table 475, 478
- IVP (installation verification procedure)
  - COBOL options 355
  - fallback 312, 347
  - installation 349
  - migration 349
  - phase 359
  - PL/I options 358
  - programs-phases relationship 350

## J

- JOB statement
  - description 56

## L

- LANGUAGE DEFAULT field of panel DSNTIPF 161
- Language Environment run time 512
- language interface token (LIT) 426
- LARGE EDM BETTER FIT field of panel DSNTIP4 172
- LE/370 LINK EDIT LIBRARY field of panel DSNTIPG 126
- LE/370 RUN TIME field of panel DSNTIPG 125
- LEVEL ID UPDATE FREQ field of panel DSNTIPL 196
- library
  - description 51
  - distribution 52
  - online xvii
  - prefix name 61
  - target 52
- LIMIT BACKOUT field of panel DSNTIPL 194
- link checker 286, 324
- LINK LIST ENTRY field of panel DSNTIPM 191
- LINK LIST LIBRARY field of panel DSNTIPT 109
- LINK LIST SEQUENCE field of panel DSNTIPM 191
- LINKNAME
  - column of USERNAMES catalog table 518
- LINKNAME column
  - IPNAMES catalog table 517
  - LOCATIONS catalog table 466, 467, 481, 516
- LIT (language interface token) 426

- LLA (LNKLST lookaside)
  - description 57
  - installation 238
  - migration 296, 333
- LMIDENT option of APPL statement 477
- LNKLST lookaside (LLA) 57
- LNKLSTxx member of SYS1.PARMLIB
  - installation 188, 238
  - migration 296, 333
- LOAD DISTRIBUTION field of panel DSNTIPT 110
- LOAD LIBRARY field of panel DSNTIPT 110
- load module library
  - DSNHDECP 242, 293, 331
  - SDSNEXIT 58
  - SDSNLINK 57
  - SDSNLOAD 57
- LOAD utility
  - SQL cursor 266
- loading
  - data
    - with an SQL cursor 266
  - distribution libraries 51
  - target libraries 51
- LOCAL DATE LENGTH field of panel DSNTIP4 169
- local deadlock cycles 181
- LOCAL TIME LENGTH field of panel DSNTIP4 169
- locale
  - definition 531
  - specifying 531
  - support for euro currency 531
- LOCALE LC\_CTYPE field of panel DSNTIPF 165
- LOCATION
  - column of LOCATIONS catalog table 480
  - column of LOCATIONS table 516
- location name
  - description 471
  - updating the BSDS 507
- lock
  - object
    - request 175
  - usage, IRLM 36
- LOCK ENTRY SIZE field of panel DSNTIPJ 182
- LOCKS PER TABLE(SPACE) field of panel DSNTIPJ 180
- LOCKS PER USER field of panel DSNTIPJ 180
- log
  - data set
    - sharing DASD 266
  - storage examples 27
- LOG APPLY STORAGE field of panel DSNTIPL 193
- log mode table 478
- logical unit (LU) 469
- LOGMODE option of MODEENT macro 479
- logon mode table entries 477
- LU 6.2 communications protocols 465
- LUNAME
  - column of LUMODES catalog table 490
  - column of LUNAMES catalog table 482
  - column of MODESELECT catalog table 491
  - field of panel DSNTIPR 466

- LUNAME (*continued*)
  - option of APPL statement
    - coding values 474
    - naming conventions 471
    - updating the BSDS 507

## M

- MACRO LIBRARY field of panel DSNTIPT 110
- main storage 39
- MANAGE THREAD STORAGE field of panel DSNTIPE 142
- MAX ABEND COUNT field of panel DSNTIPX 215
- MAX BATCH CONNECT field of panel DSNTIPE 140
- MAX DEGREE field of panel DSNTIP4 171
- MAX KEPT DYN STMTS field of panel DSNTIPE 142
- MAX REMOTE ACTIVE field of panel DSNTIPE 139
- MAX REMOTE CONNECTED field of panel DSNTIPE 140
- MAX TSO CONNECT field of panel DSNTIPE 140
- MAX TYPE 1 INACTIVE field of panel DSNTIPR 208
- MAX USERS field of panel DSNTIPE 138
- MAXBFRU option of LINE statement 502
- MAXCSA
  - setting parameter 36
- MAXDATA option of VTAM, considerations for NCP connections 503
- MAXIMUM ECSA field of panel DSNTIPJ 180
- MAXIMUM LE TOKENS field
  - panel DSNTIP7 136
- MAXPVT option of APPL statement 477
- MEMBER IDENTIFIER field of panel DSNTIPJ 181
- MEMBER NAME field of panel DSNTIPK 101
- migration
  - CLIST 7, 77
  - considerations 271, 315
  - jobs
    - description 7
    - DSNTIJEX 297, 335
    - DSNTIJIC 288, 326
    - DSNTIJIN 297, 335
    - DSNTIJMV 295, 332
    - DSNTIJTC 300, 337
    - DSNTIJUZ 293, 331
    - DSNTIJVC 290, 327
  - planning 7
  - release incompatibilities 281, 321
  - steps 10
  - tailoring session 8
- MINIMUM DIVIDE SCALE field of panel DSNTIPF 162
- mixed applications, calculating session limits 493
- mixed CCSIDs (coded character set identifiers) 526
- MIXED DATA field of panel DSNTIPF 164
- MNOTE macro error 432
- mode of VTAM
  - adding new modes 480
  - creating new modes 480
  - VTAM sessions
    - associating with sessions 488
    - default 478
    - IBMDB2LM 478

- mode of VTAM (*continued*)
  - VTAM sessions (*continued*)
    - IBMRDB 478
    - SNASVCMG 478
- mode table
  - default 475, 478
  - entering modes 477
- MODEENT macro 477, 480
- MODENAME column
  - LUMODES table 490
  - MODESELECT table 491
- MODESELECT column
  - LUNAMES catalog table 482
- MODETAB option of APPL statement 475
- modified jobs 78
- MODIFY VTAM,DEFINE command of VTAM 487
- MONITOR SIZE field of panel DSNTIPN 152
- MONITOR TRACE field of panel DSNTIPN 151
- monitoring
  - VTAM buffer pools
    - DISPLAY NET, BFRUSE 485
    - MODIFY command of MVS 485
- multi-site update
  - APPL options 476
  - MODEENT options 478
- MVS
  - defining DB2 to 235
  - IPL 253
- MVS PARMLIB updates 188

## N

- national language
  - character set identifiers 524
  - double-byte character set identifiers 525
- NCP-connected DB2s
  - considerations for MAXDATA option 503
  - sample definitions 503
- NETID 471
- NetView
  - installation 261
- NEWAUTHID column of USERNAMES catalog table 518
- notices, legal 541
- NUMBER OF COPIES field of panel DSNTIPH 104, 105
- NUMBER OF LOCK ENTRIES field of panel DSNTIPJ 182
- NUMBER OF LOGS field of panel DSNTIPL 192
- NUMBER OF TCBS field of panel DSNTIPX 215

## O

- OBJT REGISTRATION TABLE field of panel DSNTIPZ 218
- OJPERFEH migration consideration 283
- online books xvii
- OPERCNOS option of APPL statement 477
- OPTIMIZATION HINTS field of panel DSNTIP4 170
- OPTION, option of DSNMAPN macro 430

- organization application
  - DRDA access 409
  - format command 376
  - panels 395
  - with IMS 376
- OUTPUT BUFFER field of panel DSNTIPL 192
- OUTPUT MEMBER NAME field of panel DSNTIPA1 94
- overriding built-in conversion 529

## P

- pacing
  - controlling
    - description 486
    - modifying session limits 487
    - SRCVPAC option of MODEENT macro 479, 487
    - VPACING option of VTAM APPL statement 475
- PACKAGE AUTH CACHE field of panel DSNTIPP 187
- PACKAGE LISTS field of panel DSNTIPD 132
- PACKAGE STATEMENTS field of panel DSNTIPD 132
- PACKAGES field of panel DSNTIPD 132
- PAGE PROTECT field of panel DSNTIPJ 180
- panel
  - directory 81
  - ISPF 8, 77
  - organization 395
  - projects 395
- panel field names directory 82
- PARAMETER MODULE field of panel DSNTIPO 155
- PARMLIB update options 188
- PARSESS option of APPL statement 476
- password
  - VTAM 472
- PASSWORD column, USERNAMES catalog table 518
- PCTEROP option of DSNCRCT macro 446
- PERMANENT UNIT NAME field of panel DSNTIPA2 99
- phone application
  - data set format 407
  - format command 377
  - JCL 408
  - panels 406
  - program description 405
  - transaction code 380
  - using under batch 407
- PIU (path information unit)
  - description 495
  - relationship to MAXBFRU 502
- PL/I fields of panel DSNTIPG 124, 125
- PLAN
  - option of DSNCRCT macro 457
  - option of DSNMAPN macro
    - description 431
    - installation format 430
- PLAN AUTH CACHE field of panel DSNTIPP 186
- plan name for remote bind 491
- plan size, calculating 43
- PLAN STATEMENTS field of panel DSNTIPD 132
- PLANI option of DSNCRCT macro 446

PLANNAME column  
     SYSIBM.MODESELECT catalog table 491  
 PLANS field of panel DSNTIPD 132  
 PLNEXIT option of DSNCRCT macro 456  
 PLNPGME option of DSNCRCT macro 457  
 PLNPGMI option of DSNCRCT macro 447  
 PLNXTTR1 option of DSNCRCT macro 447  
 PLNXTTR2 option of DSNCRCT macro 447  
 PLT (program list table) processing 440  
 POOL THREAD TIMEOUT field of panel DSNTIP5 213  
 populating  
     CDB  
         connecting subsystems 480, 481, 516, 517  
         installation 259  
 port  
     definition 509  
     ephemeral 509  
     server 509  
     well-known 509  
 PORT  
     column of LOCATIONS table 516  
 port numbers 513  
 PPT (MVS program properties table) 235  
 prefix  
     active log 105  
     archive log 105, 198  
     library 61  
     log 105  
 PREFIX field of panel DSNTIPA1 93  
 PREPARE statement  
     migration considerations 283  
 PRIMARY QUANTITY field of panel DSNTIPA 197  
 PRIPROT option of MODEENT macro 480  
 PROC NAME field of panel DSNTIPI 176  
 program libraries 57  
 program list table (PLT) 440  
 PROGxx member of SYS1.PARMLIB 237, 296, 333  
 project application  
     format command 376  
     panels 395  
     using 399  
     with IMS 376  
 PRTCT option of APPL statement 472, 475  
 PSERVIC option of MODEENT macro 480  
 PSTOP transaction type 432  
 PURGEC option of DSNCRCT macro  
     description 447

## Q

QUIESCE PERIOD field of panel DSNTIPA 201  
 QUIESCE utility  
     migration considerations 283

## R

RACF (Resource Access Control Facility)  
     option to specify at installation or migration 184  
 RCT (resource control table)  
     defining DB2 to CICS 436  
     description 444

RCT (resource control table) (*continued*)  
     installation 253  
 RDO (resource definition online) 437  
 READ COPY2 ARCHIVE field of panel DSNTIPO 158  
 READ TAPE UNITS field of panel DSNTIPA 199  
 reading  
     online book 75  
 real storage 49  
 rebinding  
     remote package 531  
 RECALL DATABASE  
     field of panel DSNTIPO 153  
 RECALL DELAY field of panel DSNTIPO 154  
 RECEIVE job 64  
 RECORDING MAX field of panel DSNTIPA 200  
 recovery job DSNTIJDE 241  
 REGISTRATION DATABASE field of panel  
     DSNTIPZ 218  
 REGISTRATION OWNER field of panel DSNTIPZ 217  
 relational database name 471  
 RELCURHL subsystem parameter  
     migration consideration 283  
 release dependency marker 304, 342  
 release incompatibilities 281, 321  
 RELEASE LOCKS field of panel DSNTIP4 171  
 remigration 12, 312, 347  
 REMOTE LOCATION field of panel DSNTIPY 220  
 remote unit of work 465  
 REO (region error option)  
     default OPTION of DSNMAPN macro 431  
     installation 426  
 REQUIRE FULL NAMES field of panel DSNTIPZ 217  
 Resource Access Control Facility (RACF) 184  
 RESOURCE AUTHID field of panel DSNTIPP 186  
 resource control table (RCT) 253, 436  
 resource definition online (RDO) 437  
 resource limit facility (governor)  
     authorization ID 186  
     creating database during installation 259  
     specifying 154  
     storage estimation 35  
 RESOURCE TIMEOUT field of panel DSNTIPI 175  
 resource translation table (RTT) 426  
 RESTART field of panel DSNTIPA 203  
 RESYNC INTERVAL field of panel DSNTIPR 207  
 RESYNC PORT  
     field of panel DSNTIP5 511  
 RESYNC PORT field of panel DSNTIP5 210  
 RETAINED LOCK TIMEOUT field of panel  
     DSNTIPI 178  
 RETENTION PERIOD field of panel DSNTIPA 201  
 RID (record identifier) pool  
     size 42  
 RID blocks 42  
 RID POOL SIZE field of panel DSNTIPC 225  
 RLF AUTO START field of panel DSNTIPO 154  
 RLST (resource limit specification table)  
     migration considerations 273  
 RLST ACCESS ERROR field of panel DSNTIPO 154  
 RLST ACCESS ERROR field of panel DSNTIPR 206  
 RLST NAME SUFFIX field of panel DSNTIPO 154

- RO SWITCH CHKPTS field of panel DSNTIPL 195
- RO SWITCH TIME field of panel DSNTIPL 196
- ROLBE option of DSNCRCT macro 457
- ROLBI option of DSNCRCT macro 447
- rollback
  - ROLBI option of DSNCRCT macro 447
  - sync point rollback 447
- ROUTINE AUTH CACHE field of panel DSNTIPP 187
- RTT (resource translation table)
  - description 431
  - installation 426
- RUNLIB.LOAD library
  - DASD volume 99
  - device type 99
  - DSNTIJIN job 240
  - installing a second DB2 263, 265
  - naming considerations 61
- RUSIZES option of MODEENT macro 479

## S

- sample application
  - description 395
  - ODBA stored procedure 389
  - organization 401
  - output 111
  - phone 405
  - project 399
  - SQL procedure 390
  - SQL procedures processor 390
  - utilities stored procedure 387
  - verifying installation 349
- SAMPLE LIBRARY field of panel DSNTIPT 109
- sample VTAM definitions
  - description 499
  - NCP-connected DB2s 503
- scrollable cursor
  - declared temporary table for 32
- SDSNBASE library 53
- SDSNBKS library 53
- SDSNCHDR library 53
- SDSNCLST library 53
- SDSNDBRM library 53
- SDSNENU library 53
- SDSNEXIT library 53, 242
- SDSNINDX library 53
- SDSNINST library 53
- SDSNIVPD library 53
- SDSNLINK library
  - description 53, 57
  - suffix 191
- SDSNLOAD library
  - description 53
  - link list options 57
- SDSNLOD2 library 53
- SDSNMACS library 53
- SDSNSAMP library
  - description 53
  - output from panel session 78
- SDSNSHLF library 53
- SDSNSPFM library 53

- SDSNSPFP library 53
- SDSNSPFPE library 53
- SDSNSPFPK library 53
- SDSNSPFS library 53
- SDSNSPFT library 53
- SDXRRESL library 53
- SDXRSAMP library 53
- SECACPT option of APPL statement 475
- SECONDARY QTY field of panel DSNTIPA 198
- secondary server 493
- SECPROT option of MODEENT macro 480
- security
  - installation 246
  - migration 295, 332
- SECURITY\_IN column of LUNAMES catalog table 482
- SECURITY\_OUT column
  - IPNAMES catalog table 517
  - LUNAMES catalog table 482
- selection of data values 397
- SEQUENTIAL CACHE field of panel DSNTIPE 141
- server
  - DB2 as secondary server 493
- service name 510
- session 469
- session limit for VTAM
  - calculating 492
  - considerations for mixed applications 493
  - modifying default 487
  - modifying LUMODES 490
  - OPERCNOS option of APPL statement 477
  - specifying in APPL statement 474
- SHDDEST option of DSNCRCT macro 448
- sign-on exit routine
  - installation 244
  - migration 297, 335
- SIGNID option of DSNCRCT macro 448, 455
- single-byte character set identifiers 524, 526
- SIT (system initialization table) 436
- SITE TYPE field of panel DSNTIPO 158
- SMF (System Management Facility)
  - buffers 245
  - installation 244
- SMF ACCOUNTING field of panel DSNTIPN 150
- SMF STATISTICS field of panel DSNTIPN 150
- SMP/E (System Modification Program/Extended) 61
  - ACCEPT job 54, 66
  - allocation job 62, 64
  - APPLY job 54, 65
  - cleanup job 54, 65
  - copying jobs to DASD 55
  - data set options 61
  - data sets for two releases 62
  - job listings 54
  - loading DB2 libraries 51
  - RECEIVE job 54, 64
  - RELFILE format 51
  - sharing data sets with IMS 61
  - steps 9
- SNA sense code X'800A' 502, 503
- snap dump 448
- SNAP option of DSNCRCT macro 448



- SNASVCMG mode 478
- softcopy publications xvii
- SONSCIP option of APPL statement 476
- sort
  - program
    - APF authorization of library 238
- SORT LIBRARY field of panel DSNTIPW 128
- SORT POOL SIZE field of panel DSNTIPC 225
- SPUFI
  - access to remote systems 370
  - binding
    - migration 301, 339
    - remotely 370
  - testing 369
- SQL (Structured Query Language)
  - processing conversations
    - description 489
    - specifying mode 490
- SQL STRING DELIMITER field of panel DSNTIPF 162
- SQLCODE
  - 101 281
  - 538 281, 321
  - 699 281, 321
- SRBEXIT option of APPL statement 476
- SRCLIB.DATA library
  - DASD volume 99
  - device type 99
  - DSNTIJIN job 240
  - installing a second DB2 263, 265
  - naming considerations 61
- SRCVPAC option of MODEENT macro
  - for VTAM 479
  - used to control pacing 487
- SSM (subsystem member)
  - entry in IMS.PROCLIB 426
  - execution parameter 428
- SSNDPAC option of MODEENT macro
  - for VTAM 479
  - recommended value 487
- STANDBY option of DSNCRCT macro 448
- START IRLM CTRACE field of panel DSNTIPI 177
- START NAMES field of panel DSNTIPA 203
- start options for VTAM 483
- starting
  - DB2
    - installation 254
    - migration 299, 310, 336, 346
  - IRLM
    - fallback, after 310, 346
    - installation, during 254
    - migration, during 299, 336
  - TSO
    - after fallback 312, 347
    - installation 255
    - migration 300, 337
- statistics
  - report destination 448
- STATISTICS HISTORY field of panel DSNTIPO 159
- STATISTICS ROLLUP field of panel DSNTIPO 159
- STATISTICS SYNC field of panel DSNTIPN 151
- STATISTICS TIME field of panel DSNTIPN 151
- STD SQL LANGUAGE field of panel DSNTIP4 170
- STEPLIB statement of DB2 program libraries 59
- storage
  - calculating
    - external storage 25
    - main 39, 48
    - predefined models 25
    - real 49
    - virtual constraints 49
    - VTAM IOBUF 495
    - working 48
  - stored procedure
    - DB2 SQL procedures processor 268
    - DB2 SQL Procedures Processor 301, 339
    - DB2-supplied 257, 268, 269
    - DB2 UDB Control Center 268
    - DB2 UDB Control Center catalog query 268
    - DB2 UDB Control Center partition information 268
    - enabling after installation 266
    - running concurrently 38
    - SQL procedures processor 257, 269
    - subsystem parameter 257, 268
    - utilities 301, 339, 387
    - utility invocation 257, 268
    - Visual Explain 257, 268, 301, 339
- STRING DELIMITER field of panel DSNTIPF 162
- STRTWT option of DSNCRCT macro 448
- SUBID option of DSNCRCT macro 449
- subsystem
  - multiple 58, 261
  - name
    - IEFSSNxx member of SYS1.PARMLIB 236
- SUBSYSTEM MEMBER field of panel DSNTIPM 190
- SUBSYSTEM NAME field of panel DSNTIPI 175
- SUBSYSTEM NAME field of panel DSNTIPM 188
- subsystem parameter module
  - installation 242
  - migration 293, 331
  - option descriptions 104, 223, 224
- subsystem parameters
  - list 533
  - NPGTHRSH 535
  - PTASKROL 535
- SUBSYSTEM SEQUENCE field of panel DSNTIPM 190
- subtask priority 445, 456
- SUFFIX
  - option of DSNCRCT macro 449
- SUFFIX field of panel DSNTIPA1 93
- suffix identification character 449
- SUPPRESS SOFT ERRORS field of panel DSNTIPM 191
- sync point rollback 447
- SYNCLVL option of APPL statement 476
- syntax diagrams, how to read xx
- SYS1.PARMLIB library
  - updated by DSNTIJMV 235, 295, 332
- SYS1.PROCLIB library 235, 295, 332
- SYS1.VTAMLST library 483
- SYSADM authority
  - establish authorization IDs 253

- SYSADM authority (*continued*)
  - use 260
- SYSIBM.IPNAMES table of CDB
  - description 517
- SYSIBM.LOCATIONS table of CDB
  - description 480, 516
  - example 481, 517
- SYSIBM.LUMODES table of CDB
  - CONVLIMIT column 490
  - description 489
  - LUNAME column 490
  - MODENAME column 490
  - updating 490
- SYSIBM.LUNAMES table of CDB
  - description 481
  - example 483
  - specifying modes 489
  - updating 489
- SYSIBM.MODESELECT table of CDB
  - bind plan name 491
  - description 490
  - search order 491
- SYSIBM.SYSPROCEDURES catalog table migration
  - considerations 276
- SYSIBM.USERNAMES table of CDB
  - description 518
- SYSOPR authority
  - establish authorization IDs 253
- system
  - conversation
    - specifying modes 489
    - using default mode 482
- SYSTEM ADMIN 1 field of panel DSNTIPP 185
- SYSTEM ADMIN 2 field of panel DSNTIPP 185
- system initialization table (SIT) 436
- SYSTEM LOB VALUE STORAGE field of panel DSNTIP7 136
- SYSTEM MACLIB field of panel DSNTIPW 127
- System Management Facility (SMF) 244
- SYSTEM OPERATOR 1 field of panel DSNTIPP 185
- SYSTEM OPERATOR 2 field of panel DSNTIPP 185
- SYSTEM PROCEDURES field of panel DSNTIPW 127
- system services address space 37

## T

- table
  - CICS table generation procedure 444
- TABLE SPACES field of panel DSNTIPD 132
- TABLES IN STMT field of panel DSNTIPD 133
- TABLES field of panel DSNTIPD 131
- target library 52
- TASKREQ option of DSNCRCT macro 457
- TCP/IP ALREADY VERIFIED
  - field of panel DSNTIP5 511
- TCP/IP ALREADY VERIFIED field of panel DSNTIP5 211
- TCP/IP KEEPALIVE field of panel DSNTIP5 213
- TCP stack 511
- TCPALVER field of panel DSNTIP5 211
- TEMP 32K DATA SETS field of panel DSNTIPD 135

- TEMP 32K SPACE
  - field of panel DSNTIPD 255
- TEMP 32K SPACE field of panel DSNTIPD 134
- TEMP 4K DATA SETS field of panel DSNTIPD 134
- TEMP 4K SPACE
  - field of panel DSNTIPD 255
- TEMP 4K SPACE field of panel DSNTIPD 133
- TEMP CLIST LIBRARY field of panel DSNTIPT 108
- TEMP data set 78
- temporary
  - database storage estimation 30, 133
  - temporary table space, storage estimation 32
- TEMPORARY UNIT NAME field of panel DSNTIPA2 100
- TERM option
  - DSNCRCT macro 455
- THRDA option
  - DSNCRCT macro 458
- THRDM option of DSNCRCT macro 458
- THRDMAX option of DSNCRCT macro 449
- THRDS option of DSNCRCT macro 459
- thread
  - CICS
    - maximum, specifying 449, 451
    - maximum number 449
    - space 48
    - subtasks
      - priority 445, 456
    - termination
      - disconnecting 453
      - DSNCRCT macro 453
- TIME FORMAT field of panel DSNTIP4 169
- TIME TO AUTOSTART field of panel DSNTIPI 176
- TIMEOUT VALUE field of panel DSNTIPX 215
- TIMESTAMP ARCHIVES field of panel DSNTIPH 106
- TOKENE option of DSNCRCT macro 459
- TOKENI option of DSNCRCT macro 450
- TPN (transaction program name) 472
- TPN column of LOCATIONS catalog table 481
- trace
  - identification for CICS 450
- TRACE AUTO START field of panel DSNTIPN 149
- TRACE SIZE field of panel DSNTIPN 150
- TRACEID option of DSNCRCT macro 450
- TRACKER SITE field of panel DSNTIPO 158
- transaction entries 437
- transaction program name (TPN), description 472
- transmitting character data 523
- TSO
  - establishing user IDs 253
  - installation procedures 246
  - migration procedures 289, 327
  - starting 255
  - testing 360
- TSPROF option of MODEENT macro 480
- tuning
  - VTAM
    - buffer storage 485
    - creating new modes 480
    - description 484
    - MODEENT macro 487

- tuning *(continued)*
  - VTAM *(continued)*
    - pacing count 486
    - session limits 487
  - SRCVPAC option of MODEENT macro 487
- TWAIT option of DSNCRCT macro
  - TYPE=ENTRY macro 459
- TWAITI option of DSNCRCT macro 450
- two-phase commit 465
  - DRDA (Distributed Relational Database Architecture) 465
  - TCP/IP 511
  - VTAM 476
- TXID option of DSNCRCT macro 455, 460
- TXIDSO option of DSNCRCT macro
  - description 450
- TYPE
  - option of MODEENT macro 479
- TYPE column
  - USERNAMES catalog table 518
- TYPE=COMD option of DSNCRCT macro 451
- TYPE=ENTRY option of DSNCRCT macro 453
- TYPE=FINAL option of DSNCRCT macro 461
- TYPE=INIT option of DSNCRCT macro 444
- TYPE=POOL option of DSNCRCT macro 452

## U

- U LOCK FOR RR/RS field of panel DSNTIPI 177
- UCS-2 527
- Unicode
  - description 527
- UNICODE CCSID field of panel DSNTIPF 165
- UNIX System Services 513, 514, 518
- UNKNOWN AUTHID field of panel DSNTIPP 186
- UNREGISTERED DDL DEFAULT field of panel DSNTIPZ 217
- UPDATE PART KEY COLS field of panel DSNTIP4 172
- UPDATE RATE field of panel DSNTIPL 193
- updating
  - activity 400
  - communications database while DDF is active 491
  - LUMODES table of CDB 490
  - LUNAMES table of CDB 489
  - MODESELECT table of CDB 490
  - session limits 490
  - system parameters 231
- UR CHECK FREQ field of panel DSNTIPL 194
- UR LOG WRITE CHECK field of panel DSNTIPL 194
- USE FOR DYNAMICRULES field of panel DSNTIPF 166
- USE PROTECTION field of panel DSNTIPP 185
- USER
  - option of DSNCRCT macro 455
- user-defined characters 526
- USER LOB VALUE STORAGE field of panel DSNTIP7 136
- USERID option of DSNCRCT macro 456
- USERNAMES column
  - IPNAMES catalog table 517
- USERNAMES column *(continued)*
  - LUNAMES catalog table 482
- UTF-16 527
- UTF-8 527
- utilities
  - types
    - CATMAINT 300, 337
- UTILITY CACHE OPTION field of panel DSNTIPE 141
- UTILITY TIMEOUT field of panel DSNTIPI 176

## V

- VARCHAR FROM INDEX field of panel DSNTIP4 171
- VARY NET command of VTAM
  - ACTIVE option 483
- verification jobs
  - DSNTEJ0 350, 359
  - DSNTEJ1 350, 360
  - DSNTEJ1L 350, 360
  - DSNTEJ1P 350, 360
  - DSNTEJ1U 350
  - DSNTEJ2A 350, 364
  - DSNTEJ2C 350, 365
  - DSNTEJ2D 350, 351, 365
  - DSNTEJ2E 350, 351
  - DSNTEJ2F 350, 351, 366
  - DSNTEJ2P 350, 367
  - DSNTEJ2U 350, 351, 367
  - DSNTEJ3C 350, 351, 371
  - DSNTEJ3P 350, 351, 371
  - DSNTEJ4C 350, 374
  - DSNTEJ4P 350, 352, 374
  - DSNTEJ5A 350, 352, 377
  - DSNTEJ5C 350, 352, 377
  - DSNTEJ5P 350, 352, 377
  - DSNTEJ6 352, 382
  - DSNTEJ61 350, 352
  - DSNTEJ62 350, 352
  - DSNTEJ63 350, 352
  - DSNTEJ64 350, 352
  - DSNTEJ65 350, 352
  - DSNTEJ6D 352
  - DSNTEJ6P 350, 352
  - DSNTEJ6S 350, 352
  - DSNTEJ6T 352
  - DSNTEJ6U 350, 352
  - DSNTEJ6V 352
  - DSNTEJ6W 352
  - DSNTEJ6Z 352
  - DSNTEJ7 350, 353, 393
  - DSNTEJ71 350, 353
  - DSNTEJ73 350, 353
  - DSNTEJ75 350, 353
  - DSNTESA 370
  - DSNTESC 370
  - DSNTESE 370
- verification testing
  - DDF 381
  - LOB 393
- VERIFY
  - option of APPL statement 475

- IEWS field of panel DSNTIPD 131
- VIPA (virtual IP address) 511
- virtual sequential access method (VSAM) 240
- virtual storage
  - calculating constraints 49
  - calculations 39
  - layout 36
- Virtual Telecommunications Access Method (VTAM) 471
- Visual Explain
  - stored procedure 257
- VOLUME SERIAL 1-VOLUME SERIAL 7 fields of panel DSNTIPA2 98
- VPACING option of APPL statement 475
- VS COBOL fields of panel DSNTIPQ 120, 121
- VSAM (virtual storage access method)
  - clusters 240
- VTAM (Virtual Telecommunications Access Method)
  - APPL statement 473
  - buffer pools
    - calculating storage 495
    - increasing 485
    - monitoring 485
    - performance effect 485
  - commands
    - DISPLAY BFRUSE 485
    - MODIFY VTAM,DEFINE 487
    - VARY NET 483
  - defining a DB2 subsystem
    - log mode table, default 478
    - MODEENT macro 477
    - modes, default for DB2 478
    - objects needed for 473
    - sample definitions 499
  - installing 260
  - libraries
    - SYS1.SAMPLIB 478
    - SYS1.VTAMLST 483
  - parallel sessions 474
  - password 472
    - specifying in APPL statement 475
    - updating the BSDS 507
  - planning considerations 471
  - session limit 492
    - calculating 492
    - default maximum for a mode 474
  - start options 483
  - tracing 485
- VTAMFRR option of APPL statement 476

**X**

# Z

**W**

---

# Readers' Comments — We'd Like to Hear from You

DB2 Universal Database for OS/390 and z/OS  
Installation Guide  
Version 7

Publication No. GC26-9936-01

Overall, how satisfied are you with the information in this book?

|                      | Very Satisfied           | Satisfied                | Neutral                  | Dissatisfied             | Very Dissatisfied        |
|----------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| Overall satisfaction | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

How satisfied are you that the information in this book is:

|                          | Very Satisfied           | Satisfied                | Neutral                  | Dissatisfied             | Very Dissatisfied        |
|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| 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"/> |

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

---

Name

---

Address

---

Company or Organization

---

Phone No.

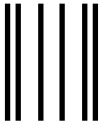


Cut or Fold  
Along Line

Fold and Tape

Please do not staple

Fold and Tape



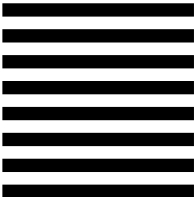
NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES

**BUSINESS REPLY MAIL**

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation  
Department HHX/H3  
PO BOX 49023  
SAN JOSE CA  
U. S. A.  
95161-9023



Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold  
Along Line





Program Number: 5675-DB2



Printed in the United States of America  
on recycled paper containing 10%  
recovered post-consumer fiber.

GC26-9936-01

