

IMS



# Installation Volume 2: System Definition and Tailoring

*Version 9*



IMS



# Installation Volume 2: System Definition and Tailoring

*Version 9*

**Note**

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 545.

**Quality Partnership Program (QPP) Edition (June 2004) (Softcopy Only)**

This QPP edition replaces or makes obsolete the previous edition, ZES1-2349-01. This edition is available in softcopy format only. The technical changes for this version are summarized under "Summary of Changes" on page xxi.

**© Copyright International Business Machines Corporation 1974, 2004. All rights reserved.**

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

# Contents

<b>Figures.</b>	xi
<b>Tables.</b>	xiii
<b>About This Book.</b>	xv
Summary of Contents	xvi
DB2 and CICS Information	xvii
Terminology and Related Publications	xvii
How to Read Syntax Diagrams.	xvii
Syntax Diagram Example	xix
How to Send Your Comments	xix
<b>Summary of Changes</b>	xxi
Changes to the Current Edition of This Book for IMS Version 9	xxi
Changes to This Book for IMS Version 9	xxi
Library Changes for IMS Version 9	xxiii
New and Revised Titles	xxiii
Terminology Changes	xxiii
Accessibility Enhancements.	xxiii
<b>Chapter 1. Designing the IMS System</b>	1
Macros and Procedures	1
Assigning System Resource Options with System Configuration Macros	2
Choosing the Number of Regions	2
Setting a Checkpoint Frequency	3
Selecting an IMS Lock Manager	4
Selecting the DL/I Separate Address Space.	4
Security Options	7
Allocating MFS Pool Space.	7
Defining Online Applications with Database and Application Macros	8
Declaring Online Databases	9
Declaring Online Programs	9
Defining IMS Transactions.	12
Defining Terminals with Data Communication Macros.	13
Defining Non-VTAM Terminals	14
Defining Switched Communication Devices	19
Specifying Macros for Multiple Systems Coupling	19
Defining VTAM Terminals	20
Specifying the Master Terminal	22
Including ETO in the IMS System	25
Including FDBR in the IMS System	26
Including RSR in the IMS System	26
<b>Chapter 2. The System Definition Process</b>	29
Selecting the Appropriate IMS System Definition	29
Using Large System Generation	30
Using DBCTL	30
Using DCCTL	31
Using the Macro Table	31
Managing System Definition	41
Sequencing Stage 1 Input.	41
LGEN System Definition Changes.	42
LGEN Stage 1 Processing.	42

LGEN Stage 2 Processing . . . . .	43
Verifying System Definition . . . . .	43
Preprocessor . . . . .	43
Executing the Preprocessor . . . . .	45
Estimating Storage Requirements for the Preprocessor . . . . .	47
Running System Definition Stage 1 . . . . .	49
Entering Stage 1 Input Statements . . . . .	49
Stage 1 and Extended Terminal Option . . . . .	50
Stage 1 Output . . . . .	50
Running System Definition Stage 2 . . . . .	51
Stage 2 Output . . . . .	51
Maintaining Stage 1 Output with SMP/E . . . . .	52
Large System Generation SMP/E JCLIN . . . . .	52
Sample JCLIN Job . . . . .	52
<b>Chapter 3. Macros . . . . .</b>	<b>55</b>
Guidelines for IMS Macros . . . . .	59
Resource Naming Rules . . . . .	60
Coding Conventions . . . . .	62
APPLCTN Macro . . . . .	62
Positional Parameters . . . . .	62
Keyword Parameters . . . . .	64
BUFPOOLS Macro . . . . .	67
Keyword Parameters . . . . .	67
COMM Macro . . . . .	70
Keyword Parameters . . . . .	71
CONFIG Macro . . . . .	78
Label Field . . . . .	79
Keyword Parameters . . . . .	79
CTLUNIT Macro . . . . .	79
Keyword Parameters . . . . .	80
DATABASE Macro . . . . .	80
Positional Parameter . . . . .	81
Keyword Parameters . . . . .	81
FPCTRL Macro . . . . .	82
Keyword Parameters . . . . .	83
IDLIST Macro . . . . .	84
Label . . . . .	84
Positional Parameters . . . . .	84
IMSCTF Macro . . . . .	85
Keyword Parameters . . . . .	85
IMSCTRL Macro . . . . .	87
Keyword Parameters . . . . .	88
IMSGEN Macro . . . . .	98
Keyword Parameters . . . . .	101
Assembler and Binder Options . . . . .	102
IMS Data Set Options . . . . .	104
Job Control Language Statements . . . . .	106
General Communication Options . . . . .	109
Security Options . . . . .	109
Sample IMSGEN Macro Statement . . . . .	110
LINE Macro . . . . .	111
Keyword Parameters . . . . .	111
LINEGRP Macro . . . . .	114
Keyword Parameters . . . . .	115
MSGQUEUE Macro . . . . .	118

Keyword Parameters . . . . .	119
MSLINK Macro . . . . .	123
Keyword Parameters . . . . .	123
MSNAME Macro . . . . .	124
Label Field . . . . .	125
Keyword Parameters . . . . .	125
MSPLINK Macro . . . . .	125
Label Field . . . . .	126
Keyword Parameters . . . . .	126
NAME Macro . . . . .	129
Positional Parameters . . . . .	131
Keyword Parameters . . . . .	133
POOL Macro . . . . .	134
RTCODE Macro . . . . .	134
Keyword Parameters . . . . .	134
SECURITY Macro . . . . .	135
Relationships between the SECURITY, COMM, and IMSGEN Macros . . . . .	136
Keyword Parameters . . . . .	137
SPAREA Macro . . . . .	140
STATION Macro . . . . .	140
Keyword Parameters . . . . .	140
SUBPOOL Macro . . . . .	141
Keyword Parameters . . . . .	142
TERMINAL Macro . . . . .	143
Label Field . . . . .	154
Keyword Parameters . . . . .	154
TRANSACT Macro . . . . .	174
Keyword Parameters . . . . .	177
TYPE Macro . . . . .	188
Keyword Parameters . . . . .	189
VTAMPOOL Macro . . . . .	190
<b>Chapter 4. Procedures . . . . .</b>	<b>191</b>
Using IMS System Definition-Supplied Procedures . . . . .	193
Storing Procedures . . . . .	193
Allocating Terminal Devices in an Online IMS System . . . . .	194
Specifying the DFSRESLB DD Statement in IMS Batch Procedures . . . . .	194
DBBBATCH Procedure . . . . .	195
DBBBATCH Procedure Description . . . . .	195
DBBBATCH Parameters . . . . .	197
DBBBATCH DD Statements . . . . .	197
DBC Procedure . . . . .	198
DBC Procedure Description . . . . .	198
DBC Parameters . . . . .	202
DBC DD Statements . . . . .	203
DBRC Procedure . . . . .	203
DBRC Procedure Description . . . . .	203
DBRC Parameters . . . . .	205
DBRC DD Statements . . . . .	205
DCC Procedure . . . . .	205
DCC Procedure Description . . . . .	205
DCC Parameters . . . . .	213
DCC DD Statements . . . . .	214
DFSJBP Procedure . . . . .	215
DFSJBP Procedure Description . . . . .	215
DFSJBP Parameters . . . . .	216

DFSJBP DD Statements . . . . .	216
DFSJMP Procedure . . . . .	217
DFSJMP Procedure Description . . . . .	217
DFSJMP Parameters . . . . .	218
DFSJMP DD Statements . . . . .	218
DFSMPR Procedure . . . . .	219
DFSMPR Procedure Description . . . . .	219
DFSMPR Parameters . . . . .	219
DFSMPR DD Statements . . . . .	220
DLIBATCH Procedure . . . . .	220
DLIBATCH Procedure Description . . . . .	220
DLIBATCH Parameters . . . . .	222
DLIBATCH DD Statements . . . . .	222
DLISAS Procedure . . . . .	222
RACF and Resource Protection . . . . .	223
DLISAS Procedure Description . . . . .	224
DLISAS Parameters . . . . .	224
DLISAS DD Statements . . . . .	224
DXRJPROC Procedure . . . . .	225
DXRJPROC Procedure Description . . . . .	225
DXRJPROC Parameter Descriptions for IRLM 2.1 . . . . .	226
DXRJPROC DD Statements for IRLM 2.1 . . . . .	226
FDR Procedure . . . . .	226
FDR Procedure Description . . . . .	226
FDR Procedure Parameters . . . . .	229
FDR Procedure DD Statements . . . . .	230
FPUTIL Procedure . . . . .	230
FPUTIL Procedure Description . . . . .	230
FPUTIL Procedure Parameters . . . . .	231
FPUTIL Procedure DD Statements . . . . .	231
IMS Procedure . . . . .	231
IMS Procedure Description . . . . .	232
IMS Procedure Parameters . . . . .	240
IMS Procedure DD Statements . . . . .	241
IMSBATCH Procedure . . . . .	242
IMSBATCH Procedure Description . . . . .	242
IMSBATCH Procedure Parameters . . . . .	243
IMSBATCH Procedure DD Statements . . . . .	243
IMSCOBGO Procedure . . . . .	244
IMSCOBGO Procedure Description . . . . .	244
IMSCOBGO Procedure Parameters . . . . .	246
IMSCOBGO Procedure DD Statements . . . . .	246
IMSCOBOL Procedure . . . . .	246
IMSCOBOL Procedure Description . . . . .	246
IMSCOBOL Procedure Parameters . . . . .	247
IMSCOBOL Procedure DD Statements . . . . .	247
IMSFP Procedure . . . . .	248
IMSFP Procedure Description . . . . .	248
IMSFP Procedure Parameters . . . . .	248
IMSFP Procedure DD Statements . . . . .	248
IMSMSG Job . . . . .	249
IMSMSG Job Description . . . . .	249
IMSPLI Procedure . . . . .	249
IMSPLI Procedure Description . . . . .	249
IMSPLI Procedure Parameters . . . . .	250
IMSPLI Procedure DD Statements . . . . .	250



	IMSPLIGO Procedure . . . . .	250
	IMSPLIGO Procedure Description . . . . .	250
	IMSPLIGO Procedure Parameters . . . . .	252
	IMSPLIGO Procedure DD Statements . . . . .	252
	IMSRDR Procedure. . . . .	252
	IMSRDR Procedure Description . . . . .	252
	IMSRDR Procedure Parameters . . . . .	253
	IMSRDR Procedure DD Statements. . . . .	253
	RDIBATCH Procedure. . . . .	253
	RDIBATCH Procedure Description . . . . .	254
	RDIBATCH Procedure Parameters . . . . .	254
	RDIBATCH Procedure DD Statements. . . . .	255
I	Parameter Descriptions for IMS Supplied Procedures . . . . .	255
	AGN= through AUTO=. . . . .	255
	BKO= through CSLG=. . . . .	259
	DBBF= through DSCT=. . . . .	264
	EMHB= through FRE=. . . . .	270
	GRNAME= through ISIS=. . . . .	274
	JVMOPMAS= through LUMP=. . . . .	279
	MAXCSA= through OVLA=. . . . .	283
	PAGES= through PWFI=. . . . .	287
	QBUF= through RVFY=. . . . .	291
	SAV= through SYS2=. . . . .	294
	TCORACF= through YEAR4=. . . . .	300
	DD Statement Descriptions . . . . .	303
I	<b>Chapter 5. Establishing a DL/I Interface from COBOL or PL/I . . . . .</b>	<b>313</b>
I	CBLTDLI. . . . .	313
I	PLITDLI . . . . .	313
I	<b>Chapter 6. Installing and Tailoring the Interactive Dump Formatter . . . . .</b>	<b>315</b>
I	Installing the Interactive Dump Formatter . . . . .	315
I	Tailoring Your Dump Formatting Options . . . . .	315
I	z/OS Dump Options . . . . .	316
I	Online System Dumps. . . . .	316
I	Online System Dependent Region Dumps . . . . .	319
I	Batch System Dumps . . . . .	319
I	IRLM Use of SDUMP . . . . .	321
	<b>Chapter 7. Installing the Transport Manager Subsystem . . . . .</b>	<b>323</b>
	VTAM Definition for the Transport Manager Subsystem . . . . .	323
	<b>Chapter 8. Tailoring the IMS System to Your Environment . . . . .</b>	<b>327</b>
	Tailoring the IMS Procedure Library . . . . .	327
	Procedures Generated by System Definition . . . . .	327
	Altering the IMS Procedure Library . . . . .	329
	Specifying IMS PROCLIB Members. . . . .	333
	Member DBFMSDBx . . . . .	333
	Member DFSCGxxx . . . . .	334
	Member DFSDCxxx . . . . .	338
	Member DFSDRFnn . . . . .	350
	Member DFSFDRxx . . . . .	351
	Member DFSFIXnn . . . . .	352
	Member DFSHSBxx . . . . .	357
	Member DFSINTxx . . . . .	361
	Member DFSJVMAP . . . . .	362

I	Member DFSJVMMS . . . . .	363
I	Member DFSJVMWK . . . . .	363
	Member DFSMPLxx . . . . .	363
	Member DFSORSxx . . . . .	365
	Member DFSPBxxx. . . . .	365
	Member DFRSRxx . . . . .	366
	Member DFSSPMxx . . . . .	377
	Member DFSSQxxx . . . . .	381
	Member DFSVSMxx . . . . .	383
	Member DFSYDTx . . . . .	413
	Member DFS62DTx . . . . .	414
	Specifying IMS Execution Parameters . . . . .	415
	Database and PSB Exec Parameters for the Control Region . . . . .	415
	Data Communication EXEC Parameters for the Control Region . . . . .	417
	System Control and Performance EXEC Parameters for the Control Region . . . . .	418
	Recovery-Related EXEC Parameters for the Control Region. . . . .	420
	Security-Related EXEC Parameters for the Control Region . . . . .	422
	IMS Message Processing Region Parameters . . . . .	422
	IMS Batch Message Processing Region Parameters . . . . .	426
	Specifying IMS Buffer Pools . . . . .	430
	VSAM Subpool Definition . . . . .	430
	OSAM Subpool Definition . . . . .	431
	OSAM Buffer Pool Compatibility Definition . . . . .	431
	Specifying VSAM and OSAM Subpools . . . . .	432
	Specifications for OSAM Sequential Buffering . . . . .	433
	Making IMS and IMSRDR Procedures Accessible to z/OS . . . . .	433
	Organizing PL/I Modules for Use with the PL/I Optimizer . . . . .	433
	Coding COBOL Programs That Run under IMS . . . . .	434
	Specifying Sequential Buffering Control Statements . . . . .	434
	Syntax of SB Control Statements. . . . .	434
	SB Parameters (SBPARAM) Control Statement . . . . .	435
	SB Image Capture (SBIC) Control Statement . . . . .	438
	SB Compare Option (SBCO) Control Statement . . . . .	438
	SB Snap (SBSNAP) Control Statement . . . . .	439
	SB Evaluation Snap (SBESNAP) Control Statement. . . . .	440
	Snap Destination (SNAPDEST) Control Statement . . . . .	441
	Specifying High-Speed Sequential Processing Control Statements . . . . .	442
	Syntax of HSSP Control Statements . . . . .	442
	Set Options (SETO) Control Statement . . . . .	442
	Set Range (SETR) Control Statement . . . . .	444
	HSSP Control Statement Examples . . . . .	445
	Supporting CCTL Users with DBCTL Databases . . . . .	446
	Preparing a CCTL . . . . .	447
	The DRA Startup Table . . . . .	447
	Managing an IMSplex Environment with a Common Service Layer . . . . .	450
	Enabling ETO Support for ACF/VTAM Terminals . . . . .	450
	ETO Descriptors . . . . .	450
	Logon Descriptors . . . . .	452
	MFS Device Descriptors . . . . .	456
	MSC Descriptors. . . . .	457
	User Descriptors . . . . .	457
	Specifying the Number of Hash Table Slots . . . . .	462
	HALDB Single Partition Processing . . . . .	463
	HALDB Processing of Single Partition Report . . . . .	463
	<b>Chapter 9. Accessing IMS Databases through the ODBA Interface . . . . .</b>	<b>465</b>

Setting Up the DRA and the ODBA Interface . . . . .	465
Creating the ODBA DRA Start-up Table . . . . .	465
Loading and Running the ODBA and DRA Modules in the z/OS Application Region . . . . .	466
Binding Application Programs . . . . .	466
Establishing and Defining Security . . . . .	466
<b>Chapter 10. Accessing IMS Databases with CICS.</b> . . . . .	471
Installing Data Sharing with CICS . . . . .	471
Installing CICS for Use with IMS Intersystem Communication . . . . .	471
Coding Considerations for PSBs . . . . .	471
Using Sequential Buffering . . . . .	472
CICS Connected to DL/I . . . . .	473
Accessing DBCTL Databases with CICS as a CCTL. . . . .	473
DBCTL Tasks . . . . .	473
CICS Tasks. . . . .	474
<b>Chapter 11. Accessing External Subsystems with IMS.</b> . . . . .	475
Installation Tasks for the IMS User . . . . .	475
Defining Your External Subsystems to IMS . . . . .	475
Defining the Language Interface Module . . . . .	481
Specifying the Desired Trace Options . . . . .	482
Specifying DB2 Groups for IMS Online Regions . . . . .	482
Specifying DB2 for IMS Batch Regions . . . . .	483
Preparing DB2 for Use with IMS . . . . .	483
Providing Appropriate Tables in Your ESS . . . . .	483
Placing DB2 Modules and Tables in the Appropriate Library . . . . .	483
<b>Appendix A. IMS System Definition (Step 3) Examples</b> . . . . .	485
IMS DB/DC Environment. . . . .	485
Data Communication Characteristics . . . . .	485
System Configuration Macro Statements . . . . .	486
Database and Application Macro Statements . . . . .	487
Data Communication Macro Statements . . . . .	490
BTAM, BSAM, GAM, and ARAM Macro Sets . . . . .	490
Switched Communication Device Macro Sets . . . . .	499
VTAM Macro Sets . . . . .	501
System Configuration Macro Statement . . . . .	510
IMS Multiple Systems Coupling . . . . .	511
Data-Sharing System Configuration . . . . .	516
Example 1. Data Sharing at the Database-Level with Update Access . . . . .	516
Example 2. Data Sharing at the Database Level with Read Access . . . . .	517
Example 3. Intra-CPC Block-Level Data Sharing . . . . .	518
Example 4. Inter-CPC Block-Level Data Sharing . . . . .	520
IMS DBCTL Environment. . . . .	521
<b>Appendix B. MFS System Definition and Programming Considerations</b> . . . . .	523
System Definition Macro Instructions . . . . .	523
BUFPOOLS Macro . . . . .	523
MSGQUEUE Macro . . . . .	523
IMSGEN Macro . . . . .	523
COMM Macro . . . . .	523
TYPE, TERMINAL, and CONFIG Macros. . . . .	524
System Programming Considerations . . . . .	526
INDEX Directory . . . . .	526
FORMAT, FORMATA, FORMATB, and TFORMAT Library Directories . . . . .	526

	Use of Concatenated Format Libraries . . . . .	526
I	Use of the Message Format Buffer pool . . . . .	527
	<b>Appendix C. Syntax Checker Example</b> . . . . .	529
I	Function Keys and the Syntax Checker . . . . .	529
	Starting the Syntax Checker . . . . .	530
I	Syntax Checker Online Help . . . . .	530
	Using the Syntax Checker . . . . .	530
	IMS Release and Control Region Panel . . . . .	531
I	IMS Release Panel . . . . .	532
	Syntax Checker Keyword Display Panel . . . . .	532
	Save Member Prompt Panel . . . . .	538
	SAVE AS Prompt Panel . . . . .	538
	Change Release Panel . . . . .	539
	Exiting the Syntax Checker . . . . .	540
	<b>Appendix D. IMS Connect Dump Formatter</b> . . . . .	541
I	Accessing the IMS Connect Dump Formatter . . . . .	542
I	IMS Version 7 Users . . . . .	542
I	Using the IMS Connect Dump Formatter . . . . .	543
	<b>Notices</b> . . . . .	545
	Trademarks. . . . .	547
	Product Names . . . . .	547
	<b>Bibliography</b> . . . . .	549
	IMS Version 9 Library . . . . .	549
	<b>Index</b> . . . . .	551

# Figures

1. Sample Code for Two 3270 Local Line Groups . . . . .	15
2. Configuration for 3270 Local Line Group . . . . .	16
3. Sample Code for a 3270 Remote Line Group . . . . .	17
4. Configuration for a 3270 Remote Line Group . . . . .	18
5. Sample Code for 3270 VTAM Terminals . . . . .	21
6. Configuration of 3270 VTAM Terminals . . . . .	22
7. Sample Code to Specify Master Terminal Configuration. . . . .	23
8. Configuration for Master Terminal. . . . .	24
9. Hierarchy of Stage 1 System Definition Macros. . . . .	42
10. JCL to Execute the Preprocessor . . . . .	46
11. Summary of the Two Stages of System Definition Processing . . . . .	49
12. Node Assignments for IMS Data Set Names . . . . .	105
13. Sample MSGEN Macro Statements . . . . .	110
14. Order of Macro Statements for Fixed and Dynamically Allocated Subpools . . . . .	130
15. Procedure to Run Offline DL/I Batch Processing Region Using IMS.ACBLIB. . . . .	196
16. Procedure to Initialize the DBRC Subordinate Address Space . . . . .	204
17. Procedure to Start a JBP Region . . . . .	216
18. Procedure to Start a JMP Region . . . . .	218
19. Procedure to Execute an IMS Message Processing Address Space. . . . .	219
20. Procedure for an Offline DL/I Batch Processing Program. . . . .	221
21. Procedure to Initialize a DL/I Separate Address Space . . . . .	224
22. Procedure to Start IRLM 2.1 . . . . .	226
23. Procedure to Execute Fast Path Utility Programs . . . . .	231
24. Procedure to Execute DB/DC or DBCTL Online Batch Message Processing Region. . . . .	243
25. Compile and Bind Procedure for IMS COBOL. . . . .	247
26. Procedure to execute a Fast Path application program . . . . .	248
27. DASD Procedure to Read IMSMSG Job. . . . .	253
28. RDIBATCH Procedure . . . . .	254
29. Transport Manager Job Control Statements . . . . .	323
30. Sample VTAM APPL Definition . . . . .	324
31. Sample VTAM MODETBL Definition . . . . .	326
32. Default Buffer Definitions . . . . .	378
33. Assembly and bind of DFSIDEF0 module . . . . .	421
34. DRA Interface to DBCTL . . . . .	447
35. System Configuration Macro Statements . . . . .	486
36. Database and Application Macro Statements . . . . .	487
37. Local SYSOUT Line Group . . . . .	491
38. Macro Statements for Local SYSOUT Line Group . . . . .	491
39. 2740 Nonswitched Line Group . . . . .	492
40. Macro Statements for 2740 Nonswitched Line Group . . . . .	492
41. 2740 Non-Station Control Line Group . . . . .	493
42. Macro Statements for 2740 Non-station Control Line Group . . . . .	493
43. 2741 Nonswitched Line Group . . . . .	493
44. Macro Statements for 2741 Nonswitched Line Group . . . . .	493
45. 3270 Local Line Groups. . . . .	494
46. Macro Statements for 3270 Local Line Group . . . . .	495
47. 3270 Remote Line Group . . . . .	496
48. Macro Statements for 3270 Remote Line Group . . . . .	497
49. System/3 Line Group . . . . .	497
50. Macro Statements for System/3 Line Group . . . . .	498
51. First System/7 Line Group . . . . .	498
52. First Macro Statements for System/7 Line Group . . . . .	498
53. Second System/7 Line Group. . . . .	499

54. Macro Statements for second System/7 Line Group . . . . .	499
55. 2740 Switched Line Group . . . . .	499
56. Macro Statements to Define 2740 Switched Line Group . . . . .	499
57. 2741 Switched Line Group . . . . .	500
58. Macro Statements to Define 2741 Switched Line Group . . . . .	500
59. 3275 Switched Line Group . . . . .	500
60. Macro Statements to Define 3275 Switched Line Group . . . . .	501
61. COMM Macro Statement . . . . .	501
62. Local 3270 VTAM Terminal. . . . .	502
63. Macro Statements to Define 3270 VTAM Terminals . . . . .	502
64. Remote 3270 VTAM Terminals . . . . .	503
65. Macro Statements to Define Remote 3270 VTAM Terminals . . . . .	503
66. Finance Communication System. . . . .	504
67. Macro Statements to Define a Finance Communication System . . . . .	505
68. Secondary Logical Unit Type 1 VTAM Terminals . . . . .	506
69. Macro Statements to Define SLU Type 1 VTAM Terminals . . . . .	506
70. NTO Devices. . . . .	507
71. Macro Statements to Define NTO Devices . . . . .	507
72. Secondary Logical Unit Type 2 VTAM Terminals . . . . .	507
73. Macro Statements to Define SLU Type 2 VTAM Terminals . . . . .	507
74. Secondary Logical Unit Type P VTAM Terminal . . . . .	508
75. Macro Statements to Define SLU Type P VTAM Terminals . . . . .	508
76. Logical Unit Type 6 VTAM Terminals . . . . .	509
77. Macro Statements to Define LU Type 6 VTAM Terminals. . . . .	510
78. System Configuration Macro Statement . . . . .	511
79. Multiple Systems Coupling . . . . .	511
80. System A IMS System Definition. . . . .	512
81. System B IMS System Definition . . . . .	513
82. System C IMS System Definition . . . . .	515
83. Data Sharing at the Database Level with Update Access. . . . .	517
84. Data sharing at the Database Level with Read Access . . . . .	518
85. Intra-CPC Block-level Data Sharing . . . . .	519
86. Inter-CPC Block-level Data Sharing . . . . .	520
87. Syntax Checker Member and Data Set Name Panel . . . . .	530
I 88. IMS Release and Control Region Panel . . . . .	531
I 89. IMS Release Panel . . . . .	532
I 90. Syntax Checker Keyword Display Panel . . . . .	533
I 91. Member Does Not Exist Panel . . . . .	537
I 92. Member is Not Processed by Syntax Checker Panel . . . . .	537
93. Save Member Prompt Panel . . . . .	538
94. SAVE AS Prompt Panel . . . . .	539
95. Change Release Panel . . . . .	540
I 96. IMS Dump Formatting Primary Menu Panel . . . . .	542
I 97. IMS-Related Product Selection Menu Panel . . . . .	543
I 98. IMS Connect Dump Formatting Menu Panel . . . . .	543
I 99. IMS Connect Dump Initialization Panel . . . . .	544
I 100. IMS Connect Dump Initialization Panel after Initialization. . . . .	544

## Tables

1.	IMS Installation Subtasks . . . . .	xv
2.	How to Read Syntax Diagrams . . . . .	xvii
3.	Changed Macros, Procedures, and PROCLIB Members for IMS Version 9 . . . . .	xxii
4.	Database and Application Macros . . . . .	8
5.	Summary of Transaction Characteristics . . . . .	12
6.	TRANSACT Macro . . . . .	12
7.	Non-VTAM Data Communication Macros . . . . .	14
8.	Switched Communication Device Macros . . . . .	19
9.	MSC Data Communication Macros . . . . .	20
10.	VTAM Data Communication Macros . . . . .	21
11.	Device Choices for Master Terminals . . . . .	23
12.	Specifying the Global Service Group (GSG) Name for RSR . . . . .	27
13.	Specifying the Transport Manager Instance (TMI) Name for RSR . . . . .	28
14.	Types of System Definition . . . . .	29
15.	Macro Table - Selecting the Appropriate IMS System Definition . . . . .	33
16.	Resource Name Table Storage. . . . .	48
17.	System Definition Macros and Their Uses. . . . .	56
18.	System Definition Macros and Their Uses. . . . .	58
19.	Maximum Occurrences of Each IMS System Definition Macro Statement . . . . .	60
20.	Calculating EPCB storage for PSB . . . . .	68
21.	Commands That Can Be Copied to the Secondary Master Terminal . . . . .	73
22.	Message Prefix Size . . . . .	120
23.	Standards to Relate Device Names to Screen Sizes. . . . .	173
24.	Relationship Between AOI= and AOI1= Specifications . . . . .	178
25.	IMS System Definition-Supplied Cataloged Procedures and the Environments in Which They Apply . . . . .	192
26.	Valid parameters for the DBBBATCH procedure . . . . .	197
27.	Valid parameters for the DLIBATCH procedure . . . . .	222
28.	Procedures for Online Systems . . . . .	261
29.	Hash Value and Resulting Lock Table Storage . . . . .	275
30.	Common Lock Table Entry values . . . . .	278
31.	Common Lock Table Entry values . . . . .	282
32.	DD Concatenations to Update . . . . .	315
33.	FMTO Options and their effect on dumps produced. . . . .	317
34.	Batch FMTO Options . . . . .	320
35.	Categories of Generated IMS Members . . . . .	327
36.	Tailoring Actions for PROCLIB Members for Batch Executions. . . . .	330
37.	Applicability of DFSCGxxx Parameters Based on Resource Manager Environment . . . . .	334
38.	RCVYCONV Values Related to SRMDEF Values . . . . .	345
39.	RCVYFP Values Related to SRMDEF Values . . . . .	345
40.	RCVYSTSN Values Related to SRMDEF Values. . . . .	345
41.	Initializing the INTERVAL Value . . . . .	359
42.	Initializing the TIMEOUT Value . . . . .	359
43.	IMS Region Types and Associated z/OS Tasks . . . . .	364
44.	Primary and Secondary Buffer Allocations . . . . .	380
45.	Trace Entries and Storage Requirements . . . . .	405
46.	Overriding Buffer Pool Sizes with Database and PSB Parameters . . . . .	416
47.	Overriding Buffer Pool Sizes with Data Communication Parameters. . . . .	417
48.	Categories and Purpose of Message Processing Region Parameters . . . . .	423
49.	Categories and Purpose of BMP Region Parameters . . . . .	426
50.	Descriptor Type Formats . . . . .	451
51.	Default Response Mode Values . . . . .	460
52.	RCVYCONV Values Related to SRMDEF Values . . . . .	460

53.	RCVYFP Values Related to SRMDEF Values . . . . .	461
54.	RCVYSTSN Values Related to SRMDEF Values. . . . .	461
I 55.	Options for Controlling Data Access for Applications that Use ODBA . . . . .	467
I 56.	Options for Defining AGN Security for Applications that Use ODBA . . . . .	467
I 57.	Options for Defining RAS Security for Applications that Use ODBA . . . . .	468
I 58.	Options for Defining APSB Security for Applications that Use ODBA . . . . .	469
I 59.	Syntax Checker Function Key Descriptions. . . . .	529
I 60.	Special Considerations for Comments in PROBLIB Members Processed by the Syntax Checker	536
I 61.	DD Concatenations . . . . .	541



## About This Book

This information is available in PDF and BookManager formats, and also as part of the IMS Version 9 QPP Information Center. To get the most current versions of the PDF and BookManager formats, go to the IMS Library page at [www.ibm.com/software/data/ims/library.html](http://www.ibm.com/software/data/ims/library.html). To get the most current versions of these books for the information center, go to the IMS V9 Vendor and Quality Partnership Program Library page at [www6.software.ibm.com/dl/ims02/imsv9lib-p](http://www6.software.ibm.com/dl/ims02/imsv9lib-p), where you can find updated plug-ins and instructions on how to install them in your IMS Version 9 QPP Information Center.

This book is one of three books that help you install and tailor an IMS Version 9 system. The other books are the *IMS Version 9: Installation Volume 1: Installation Verification* and the *IMS Version 9: Program Directory*.

This book provides detailed reference information on macros, procedures, and other system-oriented information to help you install and make modifications to IMS Version 9. (IMS Version 9 is hereafter referred to as IMS.)

As you look at the installation-related activities in Table 1, notice three key sources of information:

- Use the *IMS Version 9: Program Directory* for information on installing a new IMS system.
- Use the *IMS Version 9: Installation Volume 1: Installation Verification* after you have installed a new system to ensure that it has been installed properly.
- Use the *IMS Version 9: Installation Volume 2: System Definition and Tailoring* to tune and tailor this IMS system on an ongoing basis throughout its life.

Therefore, if you are responsible for installing a new IMS system, you should have copies of the *IMS Version 9: Program Directory* and both volumes of *IMS Installation*. If you are responsible for supporting an already-installed IMS system, you probably need access to *IMS Version 9: Installation Volume 2: System Definition and Tailoring* only.

Table 1 lists the subtasks associated with the IMS installation task and identifies the location of key information about these subtasks.

*Table 1. IMS Installation Subtasks. In this table, "Volume 1" refers to IMS Version 9: Installation Volume 1: Installation Verification and "Volume 2" refers to IMS Version 9: Installation Volume 2: System Definition and Tailoring.*

Installation Subtask	Location of Information
Installing an IMS system	<ul style="list-style-type: none"> <li>• <i>CBPDO or ServerPac documentation</i></li> <li>• <i>PSP bucket</i></li> <li>• <i>IMS Version 9: Program Directory</i></li> </ul>
Verifying the correct installation of an IMS system	<i>Volume 1</i>
Using the Installation Verification Program (IVP) system to test application or service changes	<i>Volume 1</i>
Using the IVP system for demonstrations, in-house training, or developing operation and recovery procedures	<i>Volume 1</i>
Building or moving your own systems onto a copy of the IVP system	<i>Volume 1</i>

Table 1. *IMS Installation Subtasks (continued)*. In this table, “Volume 1” refers to *IMS Version 9: Installation Volume 1: Installation Verification* and “Volume 2” refers to *IMS Version 9: Installation Volume 2: System Definition and Tailoring*.

<b>Installation Subtask</b>	<b>Location of Information</b>
Assigning system resource options with system configuration macros	<i>Volume 2</i>
Defining online applications with database and application macros	<i>Volume 2</i>
Defining terminals with Data Communication macros	<i>Volume 2</i>
Implementing ETO, RSR, or database recovery service in the IMS system	<i>Volume 2</i>
Implementing the system definition process	<i>Volume 2</i>
Installing the Transport Manager subsystem	<i>Volume 2</i>
Tailoring the IMS system for your environment	<i>Volume 2</i>
Accessing the IMS databases with CICS or DB2 database with IMS	<i>Volume 2</i>

## Summary of Contents

This book has ten chapters and three appendixes:

Chapter 1, “Designing the IMS System,” on page 1, describes the macro statements that define your system. These macro statements include system configuration macros, database and application macros, and data communication macros. This chapter also describes how to use these macros to define online applications and IMS terminals, and how to assign system resource options.

Chapter 2, “The System Definition Process,” on page 29, describes the major steps of the system definition process. These steps include selecting, managing, and verifying IMS system definition. This chapter also describes running system definition stage 1 and stage 2.

Chapter 3, “Macros,” on page 55, provides the resource naming rules, coding conventions, and syntax for all macros used in IMS system definition.

Chapter 4, “Procedures,” on page 191, identifies the cataloged procedures, control statements, and jobs that are added to PROCLIB during system definition.

Chapter 6, “Installing and Tailoring the Interactive Dump Formatter,” on page 315, provides information on IMS dumping and dump formatting options.

Chapter 7, “Installing the Transport Manager Subsystem,” on page 323, describes how to install the transport manager subsystem.

Chapter 8, “Tailoring the IMS System to Your Environment,” on page 327 shows what you can do to improve your system’s performance. The chapter also includes a database control (DBCTL) section for coordinator controller subsystem (CCTL) users.

Chapter 9, “Accessing IMS Databases through the ODBA Interface,” on page 465, describes how OS/390® application programs can access IMS DB using the Open Database Access (ODBA) API.

Chapter 10, “Accessing IMS Databases with CICS,” on page 471, explains data sharing with CICS and intersystem communications with CICS. This chapter includes a brief description of how CICS can access IMS databases using DBCTL.

Chapter 11, “Accessing External Subsystems with IMS,” on page 475, describes installation tasks required for IMS support of DB2 databases.

Appendix A, “IMS System Definition (Step 3) Examples,” on page 485, contains samples of the stage 1 input for various IMS environments.

Appendix B, “MFS System Definition and Programming Considerations,” on page 523, lists items that you should consider when planning for MFS.

| Appendix C, “Syntax Checker Example,” on page 529, describes how to use the  
 | Syntax Checker to define, verify, and validate parameters and their values in the  
 | IMS.PROCLIB members.

## DB2 and CICS Information

Only the DB2 and CICS information that applies to IMS installation is included in this manual.

For CICS customers, whenever CICS is mentioned, this book refers to CICS via DBCTL.

## Terminology and Related Publications

For a list of related publications, refer to “Bibliography” on page 549.

**Related Reading:** For definitions of terminology used in this manual and references to related information in other IMS manuals, see *IMS Version 9: Master Index and Glossary*.

---

## How to Read Syntax Diagrams

Each syntax diagram in this book begins with a double right arrow and ends with a right and left arrow pair. Lines that begin with a single right arrow are continuation lines. You read a syntax diagram from left to right and from top to bottom, following the direction of the arrows.

Table 2 describes the conventions that are used in syntax diagrams in this information:

Table 2. How to Read Syntax Diagrams








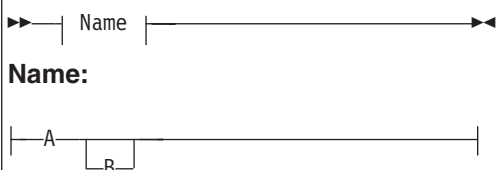

Convention	Meaning
	You must specify values A, B, and C. Required values are shown on the main path of a syntax diagram.
	You must specify value A, B, or C.
	You have the option to specify value A. Optional values are shown below the main path of a syntax diagram.

Table 2. How to Read Syntax Diagrams (continued)

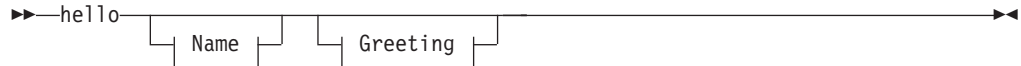
Convention	Meaning
	You have the option to specify A, B, C, or none of these values.
	You have the option to specify A, B, C, or none of these values. If you don't specify a value, A is the default.
	You have the option to specify one, more than one, or none of the values A, B, or C. Any required separator for multiple or repeated values (in this example, the comma) is shown on the arrow.
	You have the option to specify value A multiple times. The separator in this example is optional.
 <p><b>Name:</b></p> 	Sometimes a diagram must be split into fragments. The syntax fragment is shown separately from the main syntax diagram, but the contents of the fragment should be read as if they are on the main path of the diagram.
Punctuation marks and numbers	Enter punctuation marks (slashes, commas, periods, parentheses, quotation marks, equal signs) and numbers exactly as shown.
Uppercase values	Keywords, their allowable synonyms, and reserved parameters appear in uppercase letters for z/OS. Enter these values exactly as shown.
Lowercase values	Keywords, their allowable synonyms, and reserved parameters appear in lowercase letters for UNIX. Enter these values exactly as shown.
Lowercase values in italic (for example, <i>name</i> )	Supply your own text or value in place of the <i>name</i> variable.
b	A b symbol indicates one blank position.

Other syntax conventions include the following:

- When you enter commands, separate parameters and keywords by at least one blank if there is no intervening punctuation.
- Footnotes are shown by a number in parentheses, for example, (1).
- Parameters with number values end with the symbol #.
- Parameters that are names end with 'name'.
- Parameters that can be generic end with the symbol \*.

## Syntax Diagram Example

Here is an example syntax diagram that describes the `hello` command.



### Name:



### Greeting:



### Notes:

- 1 You can code up to three names.
- 2 Compose and add your own greeting (for example, how are you?).

According to the syntax diagram, these commands are all valid versions of the `hello` command:

```
hello
hello name
hello name, name
hello name, name, name
hello, your_greeting
hello name, your_greeting
hello name, name, your_greeting
hello name, name, name, your_greeting
```

The space before the `name` value is significant. If you do not code `name`, you must still code the comma before `your_greeting`.

---

## How to Send Your Comments

Your feedback is important in helping us provide the most accurate and highest quality information. If you have any comments about this or any other IMS information, you can do one of the following:

- Go to the IMS Library page at [www.ibm.com/software/data/ims/library.html](http://www.ibm.com/software/data/ims/library.html) and click the Library Feedback link, where you can enter and submit comments.
- Send your comments by e-mail to [imspubs@us.ibm.com](mailto:imspubs@us.ibm.com). Be sure to include the title, the part number of the title, the version of IMS, and, if applicable, the

specific location of the text you are commenting on (for example, a page number in the PDF or a heading in the Information Center).

---

## Summary of Changes

---

### Changes to the Current Edition of This Book for IMS Version 9

This book contains new technical information for IMS Version 9, changed technical information, and editorial changes. New information on enhancements to the System Definition process is included:

- A common execution library can be enabled for all IMS subsystem types: DB/DC with or without Fast Path, DBCTL, and DCCTL. The library is initialized by new DESC=, MCS=, and SVC2= startup parameters. See “Parameter Descriptions for IMS Supplied Procedures” on page 255 for information about the startup parameters.
- Batch environments can point to the DFSPBxxx parmlib member through the RGSUF= keyword.
- PARM1= and PARM2= parameters can be used in a batch environment. See “DBBATCH Parameters” on page 197 and “DLIBATCH Parameters” on page 222.
- The DBRC parameter is ignored on the IMSCTRL macro. See “DBRC” on page 420 for information about changing the current value for this parameter.
- A new section Appendix D, “IMS Connect Dump Formatter,” on page 541.

---

### Changes to This Book for IMS Version 9

This edition is a draft version of this book intended for use during the Quality Partnership Program (QPP). Contents of this book are preliminary and under development.

This book contains new technical information for IMS Version 9, changed technical information, and editorial changes. New information on the following enhancements is included:

- ETO feature checking enhancements
- Resource Access Security enhancements to support RACF or similar product
- Fast Path performance improvements for area open, preopen, close, and stop processing
- Support for multiple DEDB areas to share a single coupling facility structure
- Support for using multi-node persistent sessions (MNPS) to take over VTAM terminals
- Fast Path enhancements to allow optional EMHQ structure for shared queues
- Support for greater than 255 transaction scheduling classes
- Simplified address space configuration for using enhanced command environment
- IMS serviceability enhancements
- Syntax Checker enhancements to support IMS.PROCLIB members DFSPBxxx, DFSDCxxx, DFSSQxxx. See Appendix C, “Syntax Checker Example,” on page 529.
- A new section Appendix D, “IMS Connect Dump Formatter,” on page 541.

Table 3 on page xxii identifies the changed macros, procedures, and PROCLIB members related to these enhancements.

Table 3. Changed Macros, Procedures, and PROCLIB Members for IMS Version 9

Enhancement	Changes
ETO feature checking enhancements	<p><b>Macros:</b></p> <ul style="list-style-type: none"> <li>• “IMSCTRL Macro” on page 87</li> </ul>
Resource Access Security enhancements to support RACF or similar product	<p><b>Macros:</b></p> <ul style="list-style-type: none"> <li>• “SECURITY Macro” on page 135</li> <li>• “TRANSACT Macro” on page 174</li> <li>• “TYPE Macro” on page 188</li> </ul> <p><b>Procedures:</b></p> <ul style="list-style-type: none"> <li>• “IMS Procedure” on page 231</li> <li>• “DBC Procedure” on page 198</li> <li>• “DCC Procedure” on page 205</li> </ul> <p><b>IMS.PROCLIB Members:</b></p> <ul style="list-style-type: none"> <li>• “Member DFSDCxxx” on page 338</li> <li>• “Member DFSHSBxx” on page 357</li> </ul>
Fast Path performance improvements for area open, preopen, close, and stop processing	<p><b>IMS.PROCLIB Procedures:</b></p> <ul style="list-style-type: none"> <li>• “IMS Procedure” on page 231</li> </ul>
Support for multiple DEDB areas to share a single coupling facility structure	<p><b>IMS.PROCLIB Members:</b></p> <ul style="list-style-type: none"> <li>• “Member DFSVSMxx” on page 383</li> </ul>
Support for using multi-node persistent sessions (MNPS) to takeover VTAM terminals	<p><b>Macros:</b></p> <ul style="list-style-type: none"> <li>• “COMM Macro” on page 70</li> </ul> <p><b>Procedures:</b></p> <ul style="list-style-type: none"> <li>• “IMS Procedure” on page 231</li> </ul> <p><b>IMS.PROCLIB Members:</b></p> <ul style="list-style-type: none"> <li>• “Member DFSDCxxx” on page 338</li> <li>• “Member DFSHSBxx” on page 357</li> </ul>
Fast Path enhancement for optional EMHQ structure for shared queues	<p><b>IMS.PROCLIB Members:</b></p> <ul style="list-style-type: none"> <li>• “Member DFSSQxxx” on page 381</li> </ul>
Simplified address space configuration for using enhanced command environment	<p><b>IMS.PROCLIB Members:</b></p> <ul style="list-style-type: none"> <li>• “Member DFSCGxxx” on page 334</li> </ul>
Support for greater than 255 transaction scheduling classes	<p><b>Macros:</b></p> <ul style="list-style-type: none"> <li>• “TRANSACT Macro” on page 174</li> <li>• “IMSCTRL Macro” on page 87</li> </ul> <p><b>Procedures:</b></p> <ul style="list-style-type: none"> <li>• “IMS Procedure” on page 231</li> <li>• “DFSJMP Procedure” on page 217</li> <li>• “DFSMPR Parameters” on page 219</li> </ul>
IMS serviceability enhancements	<p><b>IMS.PROCLIB Members:</b></p> <ul style="list-style-type: none"> <li>• “Member DFSVSMxx” on page 383</li> </ul>



Table 3. Changed Macros, Procedures, and PROCLIB Members for IMS Version 9 (continued)

Enhancement	Changes
System Definition Enhancements	<p><b>Macros:</b></p> <ul style="list-style-type: none"> <li>• “IMSCTRL Macro” on page 87</li> </ul> <p><b>Procedures:</b></p> <ul style="list-style-type: none"> <li>• “DBBBATCH Procedure” on page 195</li> <li>• “DBC Procedure” on page 198</li> <li>• “DCC Procedure” on page 205</li> <li>• “DLIBATCH Procedure Description” on page 220</li> <li>• “FDR Procedure Description” on page 226</li> <li>• “IMS Procedure Description” on page 232</li> </ul>

## Library Changes for IMS Version 9

Changes to the IMS Library for IMS Version 9 include the addition of new titles, the change of one title, and a major terminology change. Changes are indicated by a vertical bar (|) to the left of the changed text.

### New and Revised Titles

The following list details the major changes to the IMS Version 9 library:

- *IMS Version 9: HALDB Online Reorganization Guide*

The library includes new information: *IMS Version 9: HALDB Online Reorganization Guide*. This information is available only in PDF and BookManager formats.

- *IMS Version 9: An Introduction to IMS*

The library includes new information: *IMS Version 9: An Introduction to IMS*.

- The information formerly titled *IMS Version 8: IMS Java User's Guide* is now titled *IMS Version 9: IMS Java Guide and Reference*.

- The library includes new information: *IMS Version 9: IMS Connect Guide and Reference*. This information is available only in PDF and BookManager formats.

### Terminology Changes

IMS Version 9 introduces new terminology for IMS commands:

#### type-1 command

A command, generally preceded by a leading slash character, that can be entered from any valid IMS command source. In IMS Version 8, these commands were called *classic* commands.

#### type-2 command

A command that is entered only through the OM API. Type-2 commands are more flexible and can have a broader scope than type-1 commands. In IMS Version 8, these commands were called *IMSplex* commands or *enhanced* commands.

### Accessibility Enhancements

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products. The major accessibility features in z/OS products, including IMS, enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

### **User Assistive Technologies**

Assistive technology products, such as screen readers, function with the IMS user interfaces. Consult the documentation of the assistive technology products for specific information when you use assistive technology to access these interfaces.

### **Accessible Information**

Online information for IMS Version 9 is available in BookManager format, which is an accessible format. All BookManager functions can be accessed by using a keyboard or keyboard shortcut keys. BookManager also allows you to use screen readers and other assistive technologies. The BookManager READ/MVS product is included with the z/OS base product, and the BookManager Softcopy Reader (for workstations) is available on the IMS Licensed Product Kit (CD), which you can download from the Web at [www.ibm.com](http://www.ibm.com).

### **Keyboard Navigation of the User Interface**

Users can access IMS user interfaces using TSO/E or ISPF. Refer to the *z/OS V1R1.0 TSO/E Primer*, the *z/OS V1R1.0 TSO/E User's Guide*, and the *z/OS V1R1.0 ISPF User's Guide, Volume 1*. These guides describe how to navigate each interface, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

---

## Chapter 1. Designing the IMS System

Before you can use the Information Management System Transaction Manager (IMS TM) and the Information Management System Database Manager (IMS DB), you must define the elements and functions that make up an IMS system. These include:

- Databases
- Application programs
- Terminals

IMS provides macros and procedures that enable you to define your system. The IMS macros, as well as coding and usage rules, are described in Chapter 3, “Macros,” on page 55. Procedures are described in Chapter 4, “Procedures,” on page 191.

### **In this Chapter:**

- “Macros and Procedures”
- “Assigning System Resource Options with System Configuration Macros” on page 2
- “Defining Online Applications with Database and Application Macros” on page 8
- “Defining Terminals with Data Communication Macros” on page 13
- “Including ETO in the IMS System” on page 25
- “Including FDBR in the IMS System” on page 26
- “Including RSR in the IMS System” on page 26

---

## Macros and Procedures

IMS provides a set of macro statements for system definition. You use macro statements to generate a sequence of assembler statements from a single source statement such as APPLCTN or TERMINAL. For example, to describe program resource requirements for your application programs, you use the APPLCTN macro.

IMS uses three types of macro statements to define your system:

- System configuration
- Database and application
- Data communication

Chapter 3, “Macros,” on page 55 contains all the reference information about the IMS macros. For detailed information on system-specific information, see Table 15 on page 33. Examples of macro statement sets are shown in Appendix A, “IMS System Definition (Step 3) Examples,” on page 485.

You can tailor the IMS system to suit your installation’s requirements with members of the IMS procedure library (IMS.PROCLIB). With these procedures, you can, for example:

- Initialize a DBCTL environment (DBC)
- Initialize a DB/DC environment (IMS)
- Initialize a DCCTL environment (DCC)
- Execute a Fast Path application program (IMSFP)

You can use the Syntax Checker to tailor certain PROCLIB members and to verify that the member is defined correctly.

**Related Reading:**

- For the details about the IMS procedures, see Chapter 4, “Procedures,” on page 191.
- For the details about tailoring the procedures, see Chapter 8, “Tailoring the IMS System to Your Environment,” on page 327.

---

## Assigning System Resource Options with System Configuration Macros

The system configuration macro statements describe the basic IMS control program options. The macros are as follows:

<b>Macro</b>	<b>Purpose</b>
<b>BUFPOOLS</b>	Defines the storage buffer pool sizes for the online system control program region.
<b>FPCTRL</b>	Defines the Fast Path options of the IMS control program.
<b>IMSCTF</b>	Defines additional options and system parameters under which IMS is to operate.
<b>IMSCTRL<sup>1</sup></b>	Defines the basic IMS control program options under which IMS is to operate. The ETO feature can optionally be included.
<b>IMSGEN<sup>1</sup></b>	Defines the assembler and linkage editor data sets and options, as well as the required system definition and output options.
<b>MSGQUEUE</b>	Defines the characteristics of the three user message queue data sets.
<b>SECURITY</b>	Defines the security options to be in effect during IMS execution unless they are overridden when IMS is started.

A group of parameters within the system configuration macro set relates to the integrity of the IMS online system. These parameters specify resources available for checkpoints and program isolation. The allocation of system data sets and control program storage depends upon the expected processing load and the strategy for recovery of the online IMS system.

## Choosing the Number of Regions

Overall throughput is conditionally based on the number of active dependent regions and the availability of shared resources for database processing. A processing program must have an available message region before it can enter the scheduling selection logic.

Use the MAXREGN keyword on the IMSCTRL macro to specify a working set of BMP and message regions. Up to a maximum of 999 other dependent regions can

---

1. If you are installing an IMS batch system, the IMSCTRL and IMSGEN macros are the only two that are required. IMSCTF is optional, but it contains some parameters that can be useful in your database system, and you should consider specifying it. SECURITY is also optional.

be dynamically allocated by the MTO using the /START command. Your operations procedure stipulates how many regions are active and when they are active. The scheduling algorithm controls the actual programs that execute in the regions in response to enqueued messages.

For DBCTL, the maximum sum of BMPs and DRA threads is 999.

## Setting a Checkpoint Frequency

IMS records information for restarting interrupted operations by using checkpoints. When an operation is interrupted, reprocessing occurs from the point of system interruption to a forward continuation point. Using the status information captured during checkpoint, IMS can restore the content of the message queues and database changes.

**Recommendation:** We strongly recommended that you take frequent checkpoints in order to minimize the amount of time it takes to do the restart. The trade-off for taking frequent checkpoints (and efficient restart) is an increase in processing overhead.

Begin a restart at the current checkpoint minus one. This applies to restarts from all checkpoints except RESTART, SNAPQ, and SHUTDOWN.

Use the CPLOG keyword on the IMSCTF macro to control the frequency of IMS internal checkpoints. The value you choose for this keyword represents the number of additional system log records that are created before a new checkpoint is invoked. When the increment exceeds the CPLOG value, checkpoint processing begins. The default is 1000. IMS system checkpoints are invoked by the /CHECKPOINT command or by application programs that have been authorized to enter the /CHECKPOINT command.

One technique for arriving at a CPLOG value is to use the transaction profiles and estimate, for a small peak processing interval, the number of system log events. The number of log records exceeds the total of the following:

- (Number of input transactions) x 2
- (Number of output messages) x 3
- (Number of secondary transactions) x 2
- (Number of database update calls for REPL,DLET)
- (Number of database inserts) x 2
- (Number of related index pointer maintenance calls) x 2
- (Number of programs scheduled and terminated)
- (Instances of dial lines connected and disconnected)
- (Number of checkpoint calls from programs) x 3

To insure an accurate estimate, your interval should be large enough to extend beyond the estimated response time for at least one of the longest-running transactions.

You can adjust the checkpoint frequency after observing the stability of the online IMS system. Use IMS Monitor reports to assess the processing overhead. The Region Summary report shows the total elapsed time and average elapsed time taken for checkpoints in the trace interval.

## Selecting an IMS Lock Manager

To protect database integrity, an IMS system serializes requests for database resources to ensure that no two application programs are allowed to update a database segment concurrently. Lock management is the process of controlling concurrent requests. To control the stored information about the requests, you can use either or both of the following methods:

- A program isolation lock manager
- The services of the Internal Resource Lock Manager (IRLM) component

The former only controls lock requests for a single IMS system (local locking); the latter controls requests for multiple IMS systems (global locking), as well as for a single system.

Monitoring program isolation activity is an integral part of the IMS Monitor, which is invoked by the /TRACE command. A separate program isolation trace can also be used to analyze performance. The advantage of selecting program isolation to manage locks is in not having to plan any special operating and recovery procedures, as is necessary for the IRLM.

To indicate that an IRLM is to manage locking, specify the IRLM's z/OS subsystem name on the IRLMNM parameter of the IMSCTRL macro. The IRLM is required if the IMS online system is to take part in block-level data sharing. Because the IRLM executes in a separate region, you can plan for a known amount of common storage area (ECSA) to be allocated for use by the IRLM component, or you can retain most of its control blocks in local storage rather than in the common storage area.

Your choice of lock manager is not necessarily permanent. Using parameters on the EXEC statement for the control region, you can override the use of program isolation locking. Specify IRLM=Y and give the IRLM z/OS subsystem name on the IRLMNM parameter to indicate this change. In this way, you can allow for the IMS online system to take part in block-level data sharing.

**Related Reading:** For information on administering a data-sharing environment, see *IMS Version 9: Administration Guide: System* for details on how to plan for and initialize operating procedures for an IRLM.

## Selecting the DL/I Separate Address Space

You can use a separate address space to contain DL/I code, control blocks, and buffers for full-function databases. To do this, use the local storage option (LSO). Specify LSO=S as an EXEC parameter for the control region to indicate the choice of a DL/I address space.

For the DBCTL environment, a separate DL/I address space (DLISAS) is required, and the LSO parameter is not supported.

In a DB/DC environment, if a DBCTL function is used, you must specify LSO=S.

DLISAS must also be selected for an RSR tracking subsystem that does Database Level Tracking (DLT) for full function databases.

### Storage Considerations

With the LSO=S option, storage is moved to a separate address space. Additionally, most of the PSB pool and the DMB pool are moved to the DL/I address space. These pools are not moved as a result of the LSO=Y option.

The IMS control program automatically initiates the DL/I address space. If either the control or DL/I address space terminates, the other is automatically terminated.

z/OS cross-memory services process an application database call. Because of the frequency with which cross-memory operations are used, appropriate hardware support is recommended.

IMS restart procedures are insensitive to the LSO specification. For example, an LSO=S system can be terminated, and IMS restart procedures can be performed on a system specifying LSO=Y.

The LSO=S option moves the following major storage into the DL/I separate address space:

- DL/I code
- Database buffers
- DMB pool, both resident and nonresident
- DMB work pool
- Most of the PSB pool, both resident and nonresident

For non-Fast Path systems:

- PI ENQ/DEQ tables

For an LSO=S system, the following storage is in the z/OS common area:

- OSAM code
- Log buffers
- Resident intent lists

The resident intent lists are in the control address space for all configurations other than LSO=S.

### PSB Pool Definition

In an LSO=S system, two PSB pools exist: one in the z/OS common area (DUMP) and one in the DL/I address space (DEPTHs). The DEPTHs pool contains the DL/I control blocks associated with a full-function database PCB (the JCB for example). The DUMP pool contains the communications PCBs, Fast Path PCBs, and the key feedback area of the full-function PCBs.

Specify the sizes of these pools with the SASPSB parameter on the BUFPOOLS macro. You can override these sizes with the CSAPSB and DLIPSB parameters on the IMS procedure. The command /DISPLAY P00L PSBP displays usage information for both PSB pools in an LSO=S system.

The output of the ACBGEN utility indicates, for a given PSB, the amount of space required in each pool. An out-of-space condition in the DEPTHs pool or a wait-for-storage condition is reflected in output reports of the IMS Monitor. In these reports, 'DUMP' and 'DEPTHs' represent the CSA PSB pool and the DLS PSB pool, respectively.

The ACBGEN utility organizes the PSB so that all data for the DUMP pool precedes all data for the DEPTHs pool. This organization applies to all IMS configurations.

If LSO=S is not being used, a single PSB pool is in the z/OS common area. In this case, the PSB parameter on the BUFPOOLS macro and on the IMS start procedure specifies the size of this pool.

To page fix the PSB pool, specify POOLS=DLMP and POOLS=DPSB in the DFSFIXxx member in IMS.PROCLIB. To fix the resident PSB pool, include the following module names in this member: DFSPSBRS and DFSDLIRS.

### Invoking the DL/I Address Space

IMS system definition produces the skeletal procedure DLISAS and places it in IMS.PROCLIB. You should modify this procedure as necessary and copy it to SYS1.PROCLIB.

Use the DLINM parameter on the IMSCTRL macro and the IMS start procedure to rename the procedure.

*DLISAS Procedure Modifications:* You need to plan for the following modifications to the DL/I address space procedure:

- The JCL DD statements for the full-function databases must be in the DL/I address space procedure and not in the IMS procedure. JCL DD statements for Fast Path databases and the IMS system data sets remain in the IMS procedure.
- No changes to the dynamic allocation parameter lists in IMS.SDFSRESL are required.
- The specification of the active and inactive ACBLIBs must be identical in both the IMS procedure and the DLISAS procedure. (Both the control region and the DL/I address space read ACBLIB.) The data sets used must have disposition SHR, and the concatenation order must be identical.
- IMS.PROCLIB must be defined in both the IMS and DLISAS procedures.
- The only parameters passed on the DLISAS procedure are the region type (DLS) and, optionally, the IMSID of the control program to be connected. Pool sizes and database buffering options are specified on the IMS start procedure.

*DL/I Exit Modifications:* In an LSO=S system, the DL/I exits are entered in cross-memory mode.

**Restriction:** The following operating system restrictions apply to this configuration:

- The DL/I exits cannot address storage in the control address space.
- If the DL/I exits use the IMS ISWITCH service, they must be changed to perform correctly in an LSO=S system. A macro specification of TO=DLI, not TO=CTL, must be specified on ISWITCH. The TO=DLI specification performs correctly in all IMS configurations.

### Other Planning Considerations

*Accounting Procedures with an LSO=Y System:* Accounting procedures based on the IMS system log, Resource Measurement Facility (RMF), or System Management Facility (SMF) are affected when LSO= is specified. This data reflects the existence of multiple address spaces.

The CPUTIME value recorded in the IMS system log is affected by the LSO option specified. If LSO=Y is specified, only application program processing time is recorded in the type07 log record; most DL/I processing time is not included. For some applications DL/I processing time can represent a large percentage of the total processing time. If LSO=S is specified, DL/I processing time is included in CPUTIME. Because the IMS system log does not indicate which LSO option was specified, CPUTIME values should not be used for accounting or comparison purposes when switching options.



For SMF, full-function databases are accounted for in the DL/I address space. Fast Path databases and the IMS system data sets are accounted for in the control region.

*Security Considerations:* If the full-function databases are RACF protected, the DLISAS procedure must be authorized to access these resources.

*Tuning Considerations:* You must consider that the IMS control program consists of more than one address space. You might wish to include the control region and the DL/I address space in the same RMF performance group so that RMF statistics can be compared with systems that do not use the LSO=S option.

## Security Options

The IMS system provides several ways to prevent unauthorized use of a terminal or connected device, as well as to control what processing is allowed in a dependent region. You can use application group name security to authorize the use of resources by declaring the resource name (such as the LTERM name or transaction code) in the input to the Security Maintenance utility, or you can use the resource access security protection provided by the RACF licensed program (or a similar product).

Different declarations can be made depending on the IMS security function selected. The type of security can be specified using the TYPE parameter of the SECURITY macro. For a description of the TYPE parameter, see “SECURITY Macro” on page 135.

The parameter values to invoke security checking include SIGNEXIT, TRANEXIT, and AGNEXIT. If you plan to use the RACF licensed program for your z/OS system, specify RACFTERM, RACFCOM, RACFAGN, RASRACF, or RAS. For several of the parameters, you can specify values that determine whether the MTO can override the security matrix status at restart time. These same keywords define the MTO's flexibility in invoking or overriding the type of security checking that is active during the current online execution cycle. Refer to “SECURITY Macro” on page 135 for a description of the security options available to you.

The specification of the security options needs to be a part of the overall security design.

**Related Reading:** Refer to *IMS Version 9: Administration Guide: System* for a complete description of design and operational considerations for security with IMS.

## Allocating MFS Pool Space

Several parameters of the BUFPOOLS macro allow you to reserve storage for efficient processing of communications activity. The amount of storage required primarily depends upon how much concurrent use of message format blocks is expected and how many lines are active.

### Defining Pool Space for Message Format Service (MFS) Devices

You can control the size of the message format buffer pool and the number of fetch request elements (FREs) with the BUFPOOLS macro. One FRE is required to control each active block. Without a FRE, space cannot be assigned from the message format buffer pool. When estimating the number of FREs, you should allow at least ten additional FREs to handle system activity. This eliminates the possibility of delayed response to other terminals. If preallocated FREs are not

available, a dynamic FRE is allocated from the general area of the pool. However, dynamic FREs tend to fragment the pool, and their use should be avoided whenever possible.

To determine an adequate size for the message format buffer pool, compile a list of the MFS blocks, arranging them in DIF/MID and DOF/MOD pairs. Then, record their sizes and some indication of their frequency (for example, whether they are used by priority or quick response transactions). Allow space for input/output pairs to be in the pool. Watch for large messages or those with multiple segments, because these might preempt small, frequent messages from finding space for their format blocks in the pool, and this would directly affect their response time.

**Requirement:** If you use the Message Format Service utility to build an index of the DASD addresses that reside in the message format buffer pool during online execution, the index entries also reside in extended private storage.

You can override the amount of storage for the message format buffer pool and the number of FREs using the FBP and FRE parameters in the EXEC statement for the control region.

**Related Reading:** For more information about the Message Format Service utility, see *IMS Version 9: Utilities Reference: Database and Transaction Manager*.

---

## Defining Online Applications with Database and Application Macros

IMS online applications consist of individual programs scheduled to process transactions. (Scheduling must be invoked by JCL for batch message programs.) You must identify to the control program both the available programs and the transactions that they process. You must also identify the entire set of databases that can be referred to by the programs. Use the system definition macros in Table 4 to declare your applications.

If you want data sharing to be supported, you should first understand the interaction between the IMSCTRL and DATABASE macro statements and parameters.

**Related Reading:** Before coding your data-sharing system definition, read the information on administering a data-sharing environment in *IMS Version 9: Administration Guide: System*.

Table 4 lists database and application macros, how many are coded, and the purpose of the macro.

Table 4. Database and Application Macros

Macro	Number Coded	Purpose
APPLCTN	1 per PSB	Names the application program that processes the transaction codes specified on the TRANSACT statements. The TRANSACT macro follows the APPLCTN macro in the Stage 1 input stream.
DATABASE	1 per HSAM, HISAM, HDAM, or PHDAM database; 2 per HIDAM or PHIDAM database; 1 per secondary index database that refers to another database; 1 per HSDB, DEDB	Defines all databases that IMS online control program (and DBCTL environment) manages.

Table 4. Database and Application Macros (continued)

Macro	Number Coded	Purpose
RTCODE	1 or more per APPLCTN statement	Specifies the routing codes that the input edit/routing exit routine uses to select a Fast Path application program.
TRANSACT	1 or more per APPLCTN statement; 0 for non-message BMP	Names the transaction codes to be processed by the application program specified on the APPLCTN statement. The APPLCTN macro physically precedes the TRANSACT macro in the Stage 1 input stream.

For a DBCTL environment, you define application programs only for BMP regions, using the APPLCTN macro. For an application program running in a CCTL or using the ODBA interface, use the APPLCTN macro to define the PSB names that the appropriate applications require. The TRANSACT macro is not used in defining a DBCTL system.

## Declaring Online Databases

The IMS control region allows the application program to access the database and provides DL/I call services. To declare online databases, you must define a list of all the physical databases that can be used by the programs (include DATABASE macro statements in the system definition stage 1 input). You must include separate statements for the index and data portions of a HIDAM or PHIDAM database. Also include one statement for each secondary index database that refers to any other database defined to the online system.

The name you use on the DATABASE macro is the same as that on the DBD statement (or the member name in IMS.DBDLIB) for the corresponding physical or index database definition. Do not specify logical databases.

The DATABASE macro allows you to specify a performance option for an individual database. Specify RESIDENT to load the control data management block (DMB), which is needed to access the database, from the active IMS.ACBLIBA/B data set into a separate DMB pool at system initialization. The resulting advantage of this option is that these databases do not incur OPEN or CLOSE penalties caused by competition for space in the DMB pool.

The default value (EX) of the ACCESS keyword specifies that the database is for exclusive use by this online system.

**Related Reading:** If you plan to allow concurrent use by other IMS systems, refer to “Declaring Online Databases That Share Data” in *IMS Version 9: Administration Guide: System* for an explanation of the other values for the ACCESS keyword.

## Declaring Online Programs

The online system identifies an application by a unique Program Specification Block (PSB) name, which you code in the APPLCTN macro. The message processing program uses this same name for identification. Essentially, the PSB accesses the message input queue and declares any alternative destinations, other than the input source, for messages sent by the program. The major portion of the PSB defines a complete list of database access intentions (down to the segment level, or to the

field within the segment). Before access, the PSB is prepared as a member of IMS.ACBLIB. The online system expects this control block to have been predefined.

### Declaring Program Characteristics

The APPLCTN macro allows you to specify the characteristics for application programs that run under the control of DB/DC, DBCTL, and DCCTL systems.

You decide what type of application program is being defined, either batch message or message processing. You also specify the transaction class for the messages that the program is to receive, rather than code it on subsequent TRANSACT macros.

**Requirement:** The APPLCTN macro requires that you specify whether the program can be concurrently scheduled into more than one message region or batch message region. Indicating SCHDTYP=PARALLEL permits the program to schedule into more than one region; however, this option requires that processing be truly independent. Each schedule requires its own section of any shared pools. Program isolation activity can increase if the processing has the potential to perform updates in the same database record. You cannot control the ultimate processing sequence for transactions of the same type.

On the APPLCTN macro, you can also cause the scheduling process of all environments to generate a PSB containing an I/O PCB and an alternate modifiable PCB. This generated PSB can be used, for example, by an application that makes only SQL calls. PSBGEN and ACBGEN are not required when a GPSB is utilized. You can use the GPSB= parameter to generate this type of PSB.

### Choosing PSB Performance Options

The RESIDENT option of the APPLCTN macro allows you to specify if the PSB is to be made resident during system initialization.

If the PSB is resident, the PSB is managed as follows:

- During system initialization, the total space for all resident PSBs is computed. This amount of space is obtained, and all resident PSBs are read from the active Application Control Block Library (ACBLIB) into this space.
- If SCHDTYP=PARALLEL when the program is scheduled, the resident PSB is allocated for that scheduling if it is not currently allocated to an executing program. If the resident PSB is being used, the PSB pool is searched for an inactive copy of the PSB. If one is found, that copy is allocated and used. If none is found, space for a copy is obtained in the PSB pool, and the resident PSB is copied into that space. The addresses within the copied PSB are then updated based on its new location, and its current status related to the former copy is reset.

If SCHDTYP=SERIAL when the program is scheduled, the resident copy of the PSB is allocated for that scheduling. Multiple copies of the PSB are not required; therefore, copies of the PSB are not required in the PSB pool.

Whether SCHDTYP=SERIAL or PARALLEL is specified, the PSB is read only once during initialization.

If the PSB is not resident, the PSB is managed as follows:

- When the program is scheduled, the PSB pool is searched for an inactive copy of the PSB and, if one is found, it is allocated and used. If none is found, space for a copy is obtained in the PSB pool.

If an active copy exists in the pool, it will be copied, its addresses are updated, and its status is reset to make it available for use. If no active copy exists, the PSB is read from the active ACBLIB.

When it is necessary to release a PSB in the pool, preference is given to retain “only copies” of PSBs to minimize ACBLIB I/O.

Use of the resident option eliminates I/O to ACBLIB when the PSB is scheduled. No storage fragmentation occurs in the resident PSB space as it does in the PSB pool, so more efficient use of storage is the result.

The use of the RESIDENT option should be based on the frequency with which the PSB is used. If the frequency of use would cause at least one copy of the PSB to normally be in the pool even if it were not defined as resident, it should be declared resident. If the PSB is used only occasionally, it should not be declared resident, so that its space in the PSB pool can be released when the PSB is not being used.

### Using a Dynamic PSB

The specification of the dynamic PSB option, using the dynamic option (DOPT) parameter on the APPLCTN macro, causes the PSB to be read from the active ACBLIB each time the program is scheduled. This allows an ACBGEN for a PSB to be performed while the online IMS system is running. The result of that ACBGEN can be reflected in the next scheduling of the program using that PSB.

If DOPT is specified, the PSB is managed as follows:

- No initialization for the PSB is performed at system initialization. The PSB does not need to be in the active ACBLIB during system initialization.
- When the program is scheduled, the PSB is read from the ACBLIB into the PSB pool.
- The space in the PSB pool is released when the program terminates.

DOPT can be applicable in a test configuration. It can also provide a PSB to be used with the Online Database Image Copy utility.

**Restrictions:** The following restrictions apply to a DOPT PSB:

- A dynamic PSB cannot be made resident.
- A program using a dynamic PSB cannot be scheduled in parallel.
- An MPP scheduled against a dynamic PSB cannot go through Quick Reschedule or become a pseudo WFI.
- All databases referenced by the dynamic PSB must have been defined to the system and must be present in ACBLIB at system initialization or after the last online change was performed. This is because a corresponding option does not exist for DMBs, even though the PSB is dynamic.
- The current ACBLIB must consist of two or more concatenated data sets, and the dynamic PSB must reside in any data set in the concatenation other than the first. The concatenated data sets must be of the same format and contain the output from an ACBGEN.

### Declaring Batch Message Programs

You must describe a BMP as BATCH on the PGMTYPE parameter of the APPLCTN macro. Include each BMP as a separate application with its own APPLCTN macro. You declare the BMP program by its use of a PSB, even though the batch JCL allows you to specify a program name that is different from the PSB name.

If a generalized BMP can execute with different PSBs, you must include APPLCTN macros for all of them. A choice of PSB usually involves several queues able to be processed by the one program. At execution, use the IN= parameter to match transaction queues to PSBs.

## Defining IMS Transactions

**Restriction:** This section does not apply to DBCTL.

The TRANSACT macro influences the overall online IMS system response to incoming messages. Relate one or more transaction codes to the PSB named in the prior APPLCTN macro by using the TRANSACT macro to schedule the program for execution. The transaction characteristics declared in the TRANSACT macro are summarized in Table 5.

Table 5. Summary of Transaction Characteristics

Transaction Characteristic	TRANSACT Parameter	Parameter description
What are the transaction identifiers?	CODE SYSID	Identifies unique transaction codes. For Fast Path, also declares a possible routing code. Identifies remote system and local system.
What rules govern program scheduling for this transaction?	DCLWA PARLIM PRTY	Log write-ahead option. Queue count that triggers parallel scheduling. Normal priority and alternate priority when queues build up.
What exceptions are there to the scheduling rules?	MAXRGN  PROCLIM  SCHD	Limit for number of regions that can be scheduled per transaction. Limit for number of input messages processed or the processing time. Alternative options when the transaction is unavailable.
What type of processing?	INQ FPATH MODE MSGTYPE ROUTING SERIAL SPA WFI	Update database or not. Recovery required or not. Fast Path potential transaction. Whether or not transactions are autonomous. Multisegment code or not. Response before more input or not. Whether or not program is aware of originating system. Serial processing of transaction. Conversational transaction. Wait for input transaction.
What edit operations?	EDIT	Whether or not to convert input data to uppercase. Name of an input edit routine.
What limits on output?	SEGNO SEGSIZE	Maximum segments per transaction. Maximum segment size.

From the application, you need to determine how a transaction is to be processed and how it looks to the user. Table 6 helps you specify the correct parameters on the TRANSACT macro. For a more detailed explanation of this macro, see “TRANSACT Macro” on page 174.

Table 6. TRANSACT Macro. Transaction Configurations and Recommended Parameters

Configuration	Keyword Parameter
Declares that the program should be continuously available and not reloaded each time it is scheduled.	WFI
Specifies that the transaction is an inquiry, and its database processing does not cause any updates. No recovery at system restart is desired.	INQ=(YES,NORECOV)
Declares that the transaction is an inquiry, but update processing might occur.	INQ=(NO,RECOV)

Table 6. TRANSACT Macro (continued). Transaction Configurations and Recommended Parameters

Configuration	Keyword Parameter
Declares that a response to an input message is the next event.	MSGTYPE=(RESPONSE)
Declares that the response to an input message allows multiple message input actions.	MSGTYPE=(NONRESPONSE)
Specifies that database buffers are to be written to direct access upon each request for a new message. SNGL is forced for WFI applications.	MODE=SNGL
Specifies a group of transactions that are processed together.	MODE=MULT
Declares that the transaction is to be conversational. The scratchpad area size in bytes is given.	SPA= SPA=(size in bytes)
Declares that an input message edit routine is present. The member name of the bound module in IMS.USERLB (or equivalent library) is <i>name</i> parameter. UC specifies that input is translated to uppercase before presentation to program.	EDIT=(UC, <i>name</i> )
Protects output message queues by testing the size of an output segment.	SEGSIZE= <i>size</i>
Protects output message queues by testing the number of segments issued per transaction processed.	SEGNO= <i>number</i>

### Application Program Output Limits

When an application program exceeds the output limits, a status code is returned, indicating an error. Any further attempt by the application program to exceed the limits results in abnormal termination of the program. To prevent abnormal terminations, specify values for the SEGSIZE and the SEGNO keywords on the TRANSACT macro between 0 and 65535. This causes the program to test the number and size of application program segments. This, in turn, helps prevent abnormal terminations that occur when application programs loop while inserting messages or segments into the message queues, or when they inadvertently insert segments of invalid lengths.

## Defining Terminals with Data Communication Macros

**Restriction:** This section does not apply to DBCTL.

Most of the system definition stage 1 input is made up of declarations that define terminals to be attached to the IMS online system. Sixteen macro statements describe all of your IMS system's data communication facilities. Different combinations of these statements can be used to form four macro sets:

- BTAM, BSAM, GAM, and ARAM
- Switched Communication Devices
- Multiple Systems Coupling
- VTAM

If you are using the Extended Terminal Option (ETO) of IMS TM to define terminals, you can reduce the number of macro statements required for system-defined VTAM terminals. For more information on ETO, refer to "Including ETO in the IMS System" on page 25.

An overview of each macro set and a detailed description of the macro statements in each set follow in Table 7 on page 14. Use this information and the configuration diagrams to define your terminals. Each terminal type and hardware option have equivalent parameters in one or more of the terminal-related macros in the stage 1 input.

In some instances, you must prepare the same macro statements for more than one of the sets. Refer to the section “Sequencing Stage 1 Input” on page 41 for information about the order of entering the macro sets into the assembler input stream. IMS issues stage 1 output warning messages and does not complete system definition if the macro sets are not entered in this order.

## Defining Non-VTAM Terminals

If your IMS system uses BTAM, BSAM, GAM, or ARAM, use the macro statements from this set to describe your data communication facilities. The number of sets you must prepare depends upon the hardware configuration of your IMS system.

All non-VTAM data communication specifications must precede the VTAM macro set in your IMS system definition stage 1 input deck. You receive a stage 1 output warning message if the VTAM macro set is not the last physical set. If an MSC macro set is part of your system definition, it must follow your BTAM macro set and precede your VTAM macro set; otherwise, your system definition does not complete. To add VTAM support, you must specify either an ONLINE or ALL system definition on the IMSCTRL macro statement. Non-VTAM data communication macros, the number that is coded, and their purpose are listed in Table 7.

Table 7. Non-VTAM Data Communication Macros

Macro	Number Coded	Purpose
COMM <sup>1</sup>	1 only per IMS system definition	Specifies general communication options not associated with any particular terminal type
CTLUNIT	1 for each control unit attached to the line specified by the LINE statement	Provides terminal control unit address and attributes
LINE	1 for each line following the LINEGRP data	Assigns address and characteristics of one line in a line group
LINEGRP	1 for each line type	Assigns DD names for a group of lines and terminals with like attributes
NAME	1 or more for each logical terminal or component following each TERMINAL macro	Specifies the logical terminal name for the physical terminal specified by the TERMINAL macro
STATION	1 for each System/3 and System/7	Provides System/3 or System/7 station address and attributes
TERMINAL <sup>2</sup>	1 for each physical terminal attached to the line specified by the LINE statement	Specifies physical terminal characteristics

**Note:**

1. You can specify only one COMM macro in an IMS system definition. COMM is optional for the BTAM macro set but is required for VTAM.
2. When the TERMINAL statement describes a switched physical terminal, the NAME statement cannot be used to specify logical terminal names for it.

## Specifying the Non-VTAM Terminal Configuration

An example of coding for two 3270 local line groups is shown in Figure 1 on page 15.



```
LINEGRP DDNAME=DD3270L,UNITYPE=(3270,LOCAL)          72
LINE BUFSIZE=300
  TERMINAL ADDR=310,MODEL=2
  NAME T3270L1
  TERMINAL ADDR=312,FEAT=(PFK,CARD,PEN),              X
    OPTIONS=FORCRESP,MODEL=1
  NAME T3270L2
  TERMINAL ADDR=314,TYPE=3270-A1,SIZE=(12,80)
  NAME T3270L3
  TERMINAL ADDR=316,TYPE=3270-A2,SIZE=(24,80)
  NAME T3270L4
  TERMINAL ADDR=318,TYPE=3270-A5,SIZE=(12,40)
  NAME T3270L5
  TERMINAL ADDR=31A,TYPE=3270-A6,SIZE=(6,40)
  NAME T3270L6
LINEGRP DDNAME=DD3270L2,UNITYPE=(3270,LOCAL)
LINE
  TERMINAL ADDR=311,MODEL=2,UNIT=3286
  NAME T3270P3
  TERMINAL ADDR=313,MODEL=1,UNIT=3286
  NAME T3270P4
```

*Figure 1. Sample Code for Two 3270 Local Line Groups*

The configuration for the 3270 local line group is shown in Figure 2 on page 16.



Figure 2. Configuration for 3270 Local Line Group

Figure 3 on page 17 is an example of coding for a 3270 remote line group.

```

LINEGRP DDNAME=DD3270R,UNITYPE=3270,CODE=EBCDIC          72
LINE ADDR=0C9
  TERMINAL ADDR=4040,UNIT=3275,COMPT=PTR1,                X
    FEAT=PFK,OPTIONS=COPY
    NAME T3275
    NAME T3275P,COMPT=PTR1
  CTLUNIT ADDR=C1,MODEL=2
  TERMINAL ADDR=40,FEAT=(PFK,CARD,PEN),                    X
    MODEL=2,OPTIONS=COPY
    NAME T3270A
  TERMINAL ADDR=C1,FEAT=(PFK,CARD,PEN),                    X
    MODEL=2,OPTIONS=COPY
    NAME T3270B
  TERMINAL ADDR=C2,FEAT=(PFK,CARD,PEN),                    X
    MODEL=1,OPTIONS=COPY
    NAME T3270C
  TERMINAL ADDR=C3,UNIT=3284,MODEL=2,BUFSIZE=132
  NAME T3270P1
  TERMINAL ADDR=C4,UNIT=3286,MODEL=2,BUFSIZE=132
  NAME T3270P2

```

*Figure 3. Sample Code for a 3270 Remote Line Group*

The configuration for the 3270 remote line group is shown in Figure 4 on page 18.

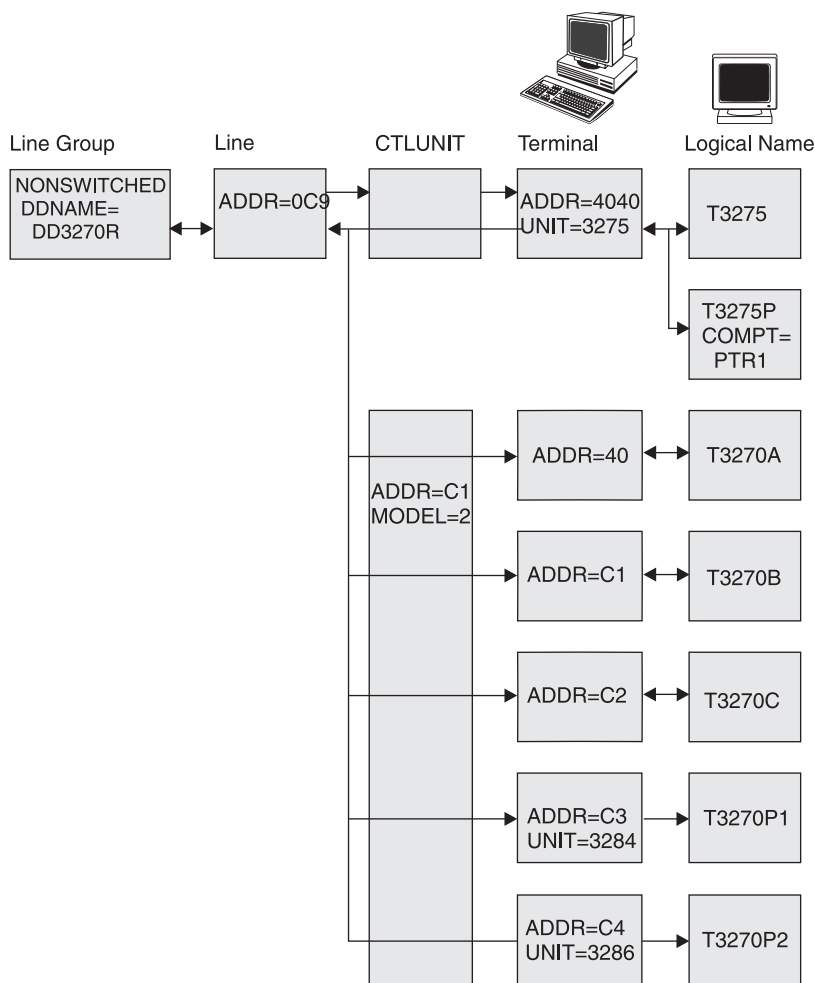


Figure 4. Configuration for a 3270 Remote Line Group

### Defining Intelligent Stations

If you plan to include remote intelligent stations in the IMS network, specify the station's physical and logical characteristics on a STATION macro. Set aside one or more line groups, and then add the characteristics of one or more System/3 or System/7 stations. An LTERM name identifies a station. IMS assigns default LTERM names as it encounters STATION macros—RSTSnnnn, where nnnn is a sequence number incremented from 0001.

The LINEGRP macro allows you to declare a System/7 line to operate either as a polled line or in contention mode. Record a two-digit hexadecimal number as the physical address of a System/7 operating on a polled line. Also declare whether the transmission is to be binary synchronous or start-stop. Optional characteristics are:

- Transmission data format (not for System/3 stations)
- Alignment and padding of data segments
- Maximum number of output blocks to the station
- No output is transmitted when the station is started or output is queued until the station issues an ASK command

Terminals attached to a System/3 or System/7 can transmit messages to IMS; however, the transmission block formats and protocols must coincide with those defined for IMS intelligent remote station support (IRSS). Each terminal can be restricted to input data only. Conversational processing and preset output destination are available.

## Defining Switched Communication Devices

If you plan to use switched 3275 devices, use the macro statements from Table 8 set to define your terminals. If prepared, these statements must be included as a subset of those prepared for the LINEGRP macro statement. Define a line group and follow this data with details of the hardware options indicated by CONFIG macro keywords. Each 3275 device carries a physical identification number as part of its circuitry. This number is made up of the last three bytes of the terminal's hardware identification and the label on the CONFIG macro. List all authorized identifications for a line group on an IDLIST macro. Subsequent LINE macros coded for 3275 devices can point to these lists. The number of sets you must prepare depends upon the hardware configuration of your IMS system. Table 8 lists macros for defining switched communication devices.

*Table 8. Switched Communication Device Macros*

Macro	Number Coded	Purpose
CONFIG	One for each type of switched 3275 configuration	Describes the configuration for a switched 3275 terminal
IDLIST	One for each switched device; must follow the LINE macro	Provides a list of hardware identifications for switched 3275 devices for a line group
LINE	One for each line of the line group	Provides the address and characteristics of one line in the line group specified by the LINEGRP statement
LINEGRP	One for each line type	Assigns DD names for a group of lines and terminals with like attributes
NAME	One or more for each logical terminal or component; must follow the SUBPOOL statement	Provides logical terminal names
POOL	One for each switched line group or one for all switched lines; must be followed by a SUBPOOL macro	Defines a pool of logical terminal names
SUBPOOL	One or more for each subpool; must follow the POOL macro	Defines a subpool of logical terminal names
TERMINAL	One for each physical terminal attached to the line specified by the LINE statement	Provides physical terminal data

## Specifying Macros for Multiple Systems Coupling

If you plan to use Multiple Systems Coupling (MSC), you must prepare one set of MSC macro statements to describe the additional data communication facilities.

Although Intersystem Communication (ISC) is part of MSC, the macro statements used to define ISC are included in the VTAM macro set and not in the MSC macro set.

**Restriction:** You must enter the macro sets in the order specified in “Sequencing Stage 1 Input” on page 41 of your IMS system stage 1 input deck. IMS issues stage 1 output warning messages and does not complete system definition if the macro sets are not entered in this order.

When you use MSC in your IMS system, the BTAM, BSAM, GAM, ARAM, and switched device macro sets (if used) must precede the MSC macro set in your IMS system definition stage 1 input deck. The VTAM macro set (if used) must occur after the MSC macro set in the stage 1 input deck. Table 9 lists macros for Multiple Systems Coupling (MSC).

Table 9. MSC Data Communication Macros

Macro	Number Coded	Purpose
MSLINK	One for each logical system-to-system link	Defines a logical link to another system
MSNAME	One for each remote and local system identification	Defines the link name block that relates to the MSLINK statement and the corresponding remote and local system identifications
MSPLINK	One for each physical link	Defines a physical link to another system in an IMS MSC configuration
NAME	One or more for each logical terminal that is to be referenced by your IMS system	Provides the logical terminal name for a physical terminal that is part of the remote system specified in the MSNAME statement

**Related Reading:** The attributes specified for MSC are described in *IMS Version 9: Administration Guide: Transaction Manager*.

## Defining VTAM Terminals

If VTAM is used by your IMS system, prepare a set of VTAM macro statements to describe the VTAM data communication facilities. The number of sets you must prepare depends upon the hardware configuration of your IMS system.

To add VTAM support, an ONLINE or ALL system definition must be specified on the IMSCTRL macro statement.

**Related Reading:** For information on the ETO options available for defining dynamic VTAM terminals, refer to the ETOFEAT keyword under “IMSCTRL Macro” on page 87.

You must enter the macro sets in the order specified in “Sequencing Stage 1 Input” on page 41 your IMS system stage 1 input deck. IMS issues stage 1 output warning messages and does not complete system definition if the macro sets are not entered in this order.

All non-VTAM data communication specifications must precede the VTAM macro set in your IMS system definition stage 1 input deck; a stage 1 output warning message occurs if the VTAM macro set is not the last physical set. If an MSC macro set is part of your system definition, it must precede the VTAM macro set, or your system definition does not complete. Table 10 on page 21 lists macros for defining VTAM terminals.

Table 10. VTAM Data Communication Macros

Macro	Number Coded	Purpose
COMM <sup>1</sup>	One only per IMS system definition	Specifies general communication options not associated with any particular terminal type
NAME	One or more for each physical terminal or component	Specifies the logical terminal name for physical terminals of the type specified by the TERMINAL or SUBPOOL statement
SUBPOOL	One for each dynamically allocated session	Defines logical unit type 6 (LU 6.1) subpools that can be dynamically allocated within the VTAMPOOL
TERMINAL	One for each terminal or component; must follow the TYPE macro	Provides the characteristics of the physical terminal type specified in the TYPE statement
TYPE	One for each terminal type	Specifies a group of VTAM terminals of the same type
VTAMPOOL	One or more per SUBPOOL; the only purpose for multiple VTAMPOOL statements is to facilitate the documentation of the intended use of the logical terminal group	Defines subpools of logical terminals to be dynamically allocated to LU 6.1 sessions

**Note:** <sup>1</sup> You can specify only one COMM macro in an IMS system definition. COMM is required for the VTAM macro set.

## Specifying the VTAM Terminal Configuration

An example of coding for a group of 3270 VTAM terminals is shown in Figure 5.

```

TYPE UNITYPE=3270,MODEL=2,PTRSIZE=132,OPTIONS=COPY
TERMINAL NAME=CT3275,UNIT=3275,COMPT=PTR1,MODEL=1
NAME VT3275
NAME VT3275P,COMPT=PTR1
TERMINAL NAME=CT3277A
NAME VT3270A
TERMINAL NAME=CT3277B
NAME VT3270B
TERMINAL NAME=CT3277C,MODEL=1
NAME VT3270C
TERMINAL NAME=CT3277D,UNIT=3284
NAME VT3270P1
TERMINAL NAME=CT3277E,UNIT=3286
NAME VT3270P2

```

Figure 5. Sample Code for 3270 VTAM Terminals

The configuration of 3270 VTAM terminals that these macros describe is shown in Figure 6 on page 22.

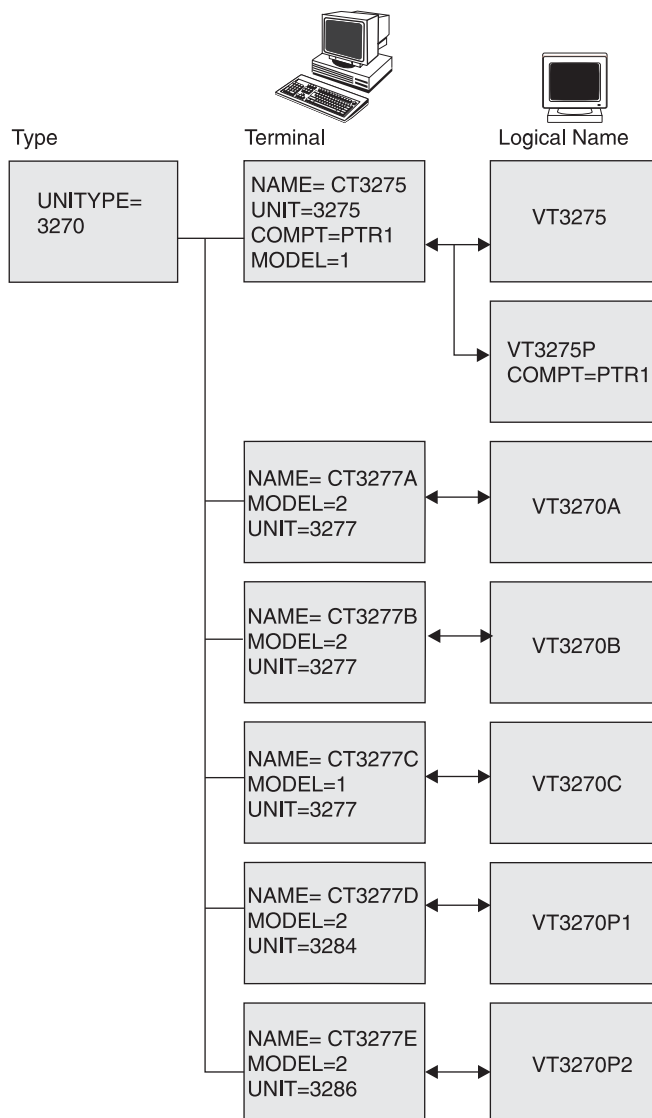


Figure 6. Configuration of 3270 VTAM Terminals

## Specifying the Master Terminal

The master terminal is the key control point for IMS online operations. Choosing a video device as the master terminal provides the advantage of convenient data entry and output response. Keep in mind, however, that you need a printed copy of many of the responses to commands, as well as a record of the system messages sent to the master terminal.

To indicate your choice for the master terminal, specify **MASTER** after the logical terminal name on the **NAME** macro. The **NAME** macro must follow the **TERMINAL** macro that describes the terminal chosen. Similarly, specify **SECONDARY** after the logical terminal name on the **NAME** macro to indicate your choice for secondary master terminal.



**Restriction:** Primary and secondary master terminals cannot be ETO or LU 6.2 device terminals. See “Member DFSDCxxx” on page 338 for additional restrictions that apply when you specify the DFSDCxxx PROCLIB member.

### Choosing Master Terminal Devices

Some restrictions apply for choosing the device for master and secondary terminals, as shown in Table 11. Your choice for secondary master terminal depends on the expected amount of output and the user’s requirement for promptness of printing.

Table 11. Device Choices for Master Terminals

Master Terminal	Choice of Device Type
Primary	3270 <sup>1</sup>
	SLUTYPE1 (console)
	SLUTYPE2 (console) <sup>2</sup>
	2740 <sup>3</sup>
Secondary	328x <sup>1</sup>
	SLUTYPE1 (first component)
	2740 <sup>3</sup>
	SPOOL

#### Notes:

<sup>1</sup> If a 3270 is specified as the master terminal, a 328x or SPOOL device must be specified as secondary.

<sup>2</sup> SLUTYPE2 requires a secondary master terminal. The NAME macro for the secondary must be placed before of the primary terminal.

<sup>3</sup> The 2740 must have station control.

The /ASSIGN command can be used to switch the secondary master console to another destination such as a spool SYSOUT line group.

### Specifying the Master Terminal Configuration

An example of coding for a master terminal on a 3270 local line is shown in Figure 7.

```

TYPE UNITYPE=(3270,LOCAL),MODEL=2
TERMINAL NAME=L3270A
NAME (VT3270L1,MASTER)
TERMINAL NAME=L3270B,MODEL=1,OPTIONS=FORCRESP
NAME VT3270L2
TERMINAL NAME=L3270C,TYPE=3270-A2,SIZE=(24,80)
NAME VT3270L3
TERMINAL NAME=L3270D,TYPE=3270-A3,SIZE=(32,80)
NAME VT3270L4
TERMINAL NAME=L3270E,TYPE=3270-A4,SIZE=(43,80)
NAME VT3270L5
TERMINAL NAME=L3284A,UNIT=3284,PTRSIZE=132
NAME (VT3270P3,SECONDARY)

```

Figure 7. Sample Code to Specify Master Terminal Configuration

The configuration for the master terminal is shown in Figure 8 on page 24:

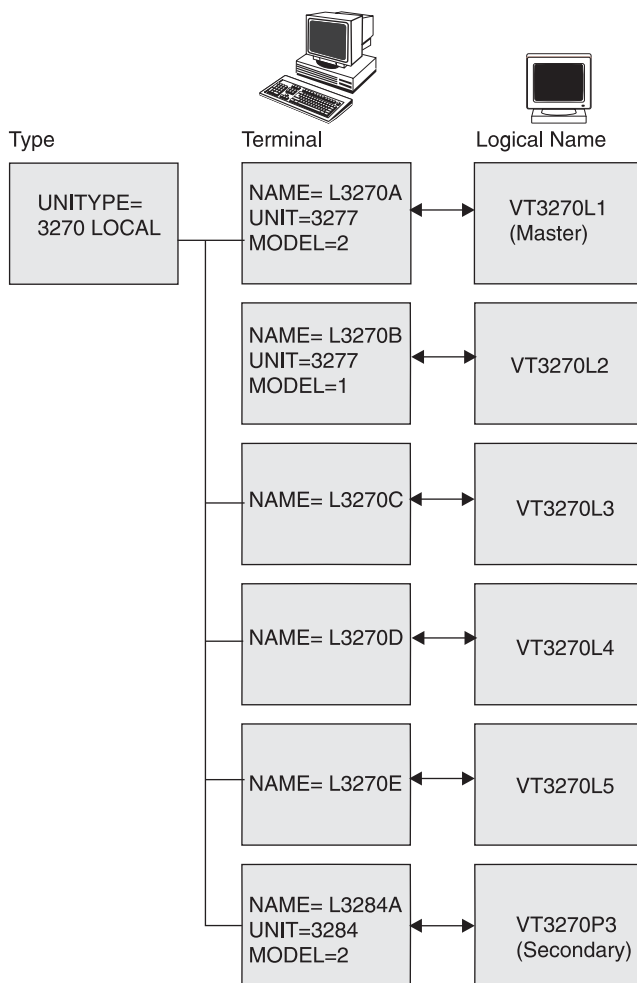


Figure 8. Configuration for Master Terminal

### Choosing the Extent of Secondary Master Logging

To provide for automatic copying of the entry and response of key system control commands, specify values for the COPYLOG keyword on the COMM macro. These values cause all command activity issued by any terminal to be copied.

**Recommendation:** It is recommended that you specify ALL, unless terminals other than the master terminal are likely to issue many commands. The choice of NONE or NOMASTER is not recommended, because the printed log of activity provides a valuable audit mechanism. Your choice of automatic copying does not affect the logging of IMS system messages to the secondary master terminal.

The commands that can be copied to the secondary master terminal are provided under the COMM macro description.

---

## Including ETO in the IMS System

The Extended Terminal Option (ETO) of IMS TM enables VTAM terminals to log on to IMS TM, even if they are not defined during the IMS system definition process. IMS TM dynamically builds the required control blocks and queues based on VTAM information and IMS skeleton definitions called descriptors. ETO is an optional feature; its installation is verified during IMS TM initialization.

**Definitions:** Terminals defined through a system definition are *static* terminals. Terminals not defined through a system definition are *dynamic* terminals.

The ETO feature applies to all VTAM terminals except for MSC VTAM, MTO terminals, and XRF surveillance links. These, and terminals supported by non-VTAM access methods, still require an IMS system definition to introduce changes. The number of terminals that you define dynamically is limited only by your resource constraints. With ETO, you can:

- Improve your system availability by reducing scheduled down time associated with adding or deleting VTAM terminals
- Improve IMS security by relating output message queues (LTERMs) to users rather than to terminals
- Reduce the number of macro statements required for static, system-defined VTAM terminals
- Reduce virtual storage use for ETO terminals and users by allocating storage only when it is actually required

**Restriction:** Without ETO, adding, deleting, or changing terminals defined to IMS TM requires that your online system be terminated down to incorporate the changes. Additionally, the stage 1 input can become large if you are defining a large number of terminals.

With ETO, IMS TM does not need to be terminated and cold started to add new terminals and users. ETO descriptors are not generated during LGEN; however, a standard system definition can build ETO descriptors that support terminals that are currently generated statically. This allows an easy transition to the ETO environment.

Dynamic allocation of VTAM terminals does not require the use of VTAM TYPE, TERMINAL, NAME, VTAMPOOL, or SUBPOOL macros, or the use of MSC remote NAME macros, for the system definition stage 1 input. Instead, you include ETO with the ETOFEAT keyword on the IMSCTRL macro, which causes IMS TM to generate descriptors. IMS TM uses descriptors to build the required control blocks and queues associated with terminal definition. With ETO terminals, configuration information from VTAM control blocks is dynamically merged with information from ETO descriptors when a user logs onto IMS TM. Terminal control blocks are not built for ETO terminals until an ACF/VTAM session is established between the terminal and IMS TM, or until a user structure is built. User control blocks are not built until a user signs on to a terminal.

You can continue to define ACF/VTAM terminals statically.

**Requirement:** Static terminal definition is still required for:

- BTAM (non-VTAM) terminals
- MTO and secondary master terminals
- MSC physical/logical links

XRF ISC Surveillance link

ETO descriptors are described in “Enabling ETO Support for ACF/VTAM Terminals” on page 450.

**Related Reading:** For information on administering ETO, see *IMS Version 9: Administration Guide: Transaction Manager*. For information on security for ETO, see *IMS Version 9: Administration Guide: System*.

---

## Including FDBR in the IMS System

Fast Database Recovery (FDBR) provides quick access to shared database resources in a sysplex environment that might otherwise be locked by a failed IMS until the failed system is restarted.

You can include Fast DB Recovery (FDBR) for DB/DC or DBCTL subsystems.

- To enable a DB/DC subsystem for FDBR:

Specify the FDRMBR parameter in the IMS procedure. The FDRMBR parameter defines the DB/DC system as FDBR-capable.

**Restriction:** If both FDRMBR and HSBID (XRF configuration) parameters are specified in the IMS procedure, the FDRMBR parameter is ignored.

You must supply the IMS ID of the system that FDBR is to track. You specify this IMS ID in the control statement for the DFSFDRxx IMS.PROCLIB member, and it must match the ID you specified for the IMSID EXEC parameter.

- To enable a DBCTL subsystem for FDBR:

Specify the FDRMBR parameter in the DBC procedure. The FDRMBR parameter defines the DBCTL system as FDBR-capable. You should make note of the IMS ID of the system FDBR will track. You specify this IMS ID in the control statement for the DFSFDRxx IMS.PROCLIB member, and it must match the ID you specify for the IMSID EXEC parameter.

**Restriction:** If you specify both FDRMBR and DBRSE (DBCTL standby configuration) parameters in the DBC procedure, the FDRMBR parameter is ignored.

---

## Including RSR in the IMS System

Remote Site Recovery (RSR) enables you to recover quickly from an interruption of computer services at an active (primary) site. RSR supports recovery of IMS DB full-function databases, IMS DB Fast Path DEDBs, IMS TM message queues, and the IMS TM telecommunications network.

The system definition process for RSR involves setting up a global service group (GSG) and a transport manager instance (TMI) both of which can be defined in several different places. This section describes how the various parts of RSR system definition fit together.

You must specify RSRFEAT=RLT or RSRFEAT=DLT in your IMSCTRL macro to enable RSR. You also need to specify a GSG name and a TMI name.

Additional RSR-related parameters and overrides are specified in a DFSRSRxx member that you must supply in IMS.PROCLIB.

If the RSR feature is installed and you do not want to use it, you must override RSR enablement by specifying RSR(NO) in the DFSRSRxx proclib member. For more information, see "Member DFSRSRxx" on page 366.

As the example in Table 12 shows, you can specify a GSG name in several places: the IMSCTRL macro, the DLIBATCH procedure, the DBBBATCH procedure, the DFSRSRxx PROCLIB member, and on certain DBRC commands. Table 13 on page 28 shows where you can specify a TMI name: the IMSCTRL macro, the DLIBATCH and DBBBATCH procedures, the DFSRSRxx PROCLIB member, and also on certain transport manager and isolated log sender commands.

To define defaults for the GSG or TMI names, specify the GSGNAME= or TMINAME= statement in the IMSCTRL macro. To override the specification in the IMSCTRL macro, specify the:

- GSGNAME= and TMINAME= statements in the DLIBATCH or DBBBATCH procedure
- GSGNAME and TMINAME parameters in the DFSRSRxx PROCLIB member

After the GSG name and TMI name have been set or overridden, you can use these names in various DBRC and transport manager and isolated log sender commands.

**Example:** If your GSG name is payroll for your payroll system and your TMI name is ELX1, these are the DBRC, TMS, and ILS commands:

- DBRC commands:
 

```
INIT.GSG GSGNAME(accounts)
INIT.DB DBD(db1) GSGNAME(payroll)
CHANGE.DB DBD(db1) GSGNAME(payroll)
INIT.DBDS DBD(dedb1) AREA(area1) GSGNAME(payroll)
CHANGE.DBDS DBD(dedb1) AREA(area1) GSGNAME(payroll)
```
- TMS and ILS commands
 

```
SET INSTANCE(ELX1)
START ILS(payroll)
```

Table 12 explains how to specify the Global Service Group (GSG) name for RSR. For this example, assume accounts is the GSG name.

Table 12. Specifying the Global Service Group (GSG) Name for RSR

Procedure	Result
Specify <b>GSGNAME=accounts</b> in the IMSCTRL macro.	Default name is specified.
Specify GSGNAME statement, <i>GSG=payroll</i> , in the DBBBATCH or DLIBATCH procedure. This name <i>must match</i> GSGNAME parameters.	Default name ( <b>accounts</b> ) is overridden.
Specify the GSGNAME parameter(s), <i>GSGNAME (payroll)</i> , in the DFSRSRxx PROCLIB member. Parameters <i>must match</i> the statement in the DBBBATCH or DLIBATCH procedure.	Default parameters are overridden.

Table 12. Specifying the Global Service Group (GSG) Name for RSR (continued)

Procedure	Result
Enter DBRC commands using the GSG name, <i>payroll</i> , or the default name, <b>account</b> : GSGNAME (payroll) INIT.DB INIT.DBDS GSGNAME(payroll) CHANGE.DB CHANGE.DBDS GSGNAME(payroll)	Using override name(s) generates same results as using default name.
Enter ILS command(s) START ILS(payroll)	Using override name(s) generates same results as using default name.

In Table 13, ELX1 is an example of a TMI name. You can use the default or overridden TMI name when you issue the TMS command.

Table 13. Specifying the Transport Manager Instance (TMI) Name for RSR

Procedure	Result
Specify <b>UNITTYP=3270 LOCAL</b> in the IMSCTRL macro.	Default name is specified.
Specify TMINAME= statement, <i>TMI=ELX1</i> , in the DBBBATCH or DLIBATCH procedure. This name <i>must match</i> TMINAME parameter.	Default parameter is overridden.
Specify the TMI parameter <i>TMI=ELX1</i> in the DFSRSRxx PROCLIB member. Parameter <i>must match</i> statement in DBBBATCH or DLIBATCH procedure.	Default parameter is overridden.
Enter TMS commands using the TMI name, <i>ELX1</i> : SET INSTANCE(ELX1)	Using override name generates same results as using default name.

A single active online subsystem environment does not require a separate system definition for the tracking subsystem. A separate tracking subsystem system definition is required if:

- More than one active subsystem must be tracked, and no one active subsystem system definition defines all of the resources needed by the tracker.  
This separate system definition requirement can be eliminated if the system definition of one of the active subsystems is enlarged to define the additional resources.
- Batch DL/I jobs use databases that are not defined in the active subsystem.
- You want to eliminate database definitions for databases that are not tracked from the tracking subsystem.
- You want to reduce the data communications definitions for the tracking subsystem.

## Chapter 2. The System Definition Process

The system definition process does not build a complete system. Some elements are built outside of the system definition process. You can create JCL to build these elements by running the SMP/E GENERATE command.

**Related Reading:** For more information about using the SMP/E GENERATE command, see *IMS Version 9: Release Planning Guide*.

### In this Chapter:

- “Selecting the Appropriate IMS System Definition”
- “Managing System Definition” on page 41
- “Verifying System Definition” on page 43
- “Running System Definition Stage 1” on page 49
- “Running System Definition Stage 2” on page 51
- “Maintaining Stage 1 Output with SMP/E” on page 52

System definition examples for a variety of IMS environments are provided in Appendix A, “IMS System Definition (Step 3) Examples,” on page 485. This appendix also provides examples of data communication macro statements and data sharing system configurations.

## Selecting the Appropriate IMS System Definition

The IMS system definition includes CLASS and TYPE. The types are: ALL, BATCH, CTLBLKS, MODBLKS, MSVERIFY, NUCLEUS, and ON-LINE. The three system definition environment classes are: DB/DC, DBCTL, and DCCTL. DB/DC is the system definition class that builds a standard IMS system. The DBCTL class builds only the DBCTL environment. The DCCTL class builds only the DCCTL environment.

Table 14 describes the types of system definition specified by the IMSCTRL macro, their typical use, and the results.

Table 14. Types of System Definition

IMSCTRL Option	When Used	Result
ALL	Typical initial system definition. Not usually needed for maintenance.	Builds most IMS libraries. Includes BATCH and ONLINE options.
BATCH	Only for the batch environment.	Builds batch environment libraries.
CTLBLKS	Convenience update. Includes bind of existing nucleus with same suffix.	Control block definition. Includes MSVERIFY and MODBLKS options.
MODBLKS	Convenience update. Used for the online change option. <b>Note:</b> If you run stage 1 and stage 2 processing with the IMSCTRL macro without specifying MODBLKS, a cold start is required. Failing to specify MODBLKS causes updates to IMS.SDFSRESL, which requires the new IMS.SDFSRESL to be copied and, therefore, a cold start is necessary.	Builds control blocks that enable database, program, transaction, and Fast Path routing code changes to be implemented during online operation.

Table 14. Types of System Definition (continued)

IMSCTRL Option	When Used	Result
MSVERIFY	Only appropriate for MSC.	Builds control blocks for the MSC Verification utility.
NUCLEUS	Major maintenance.	Builds IMS nucleus and control blocks. Includes CTLBLKS option.
ON-LINE	Major update or initial system definition. Often required for maintenance.	Builds most IMS libraries. Includes all but BATCH option.

**Requirement:** After your initial system definition, the ON-LINE, CTLBLKS, and NUCLEUS types of system definition are used to implement most changes. These system definitions require a cold start of the IMS online system to take effect.

However, for certain modifications and additions, you can take advantage of the online change method using the MODBLKS system definition. The changes are made active during the execution of the online system and do not require a restart operation.

**Related Reading:** For more information on managing online change, refer to *IMS Version 9: Administration Guide: System*.

## Using Large System Generation

LGEN is a system definition configuration, not a system definition type. Use this configuration when stage 1 or stage 2 job steps require more than 4 megabytes of private storage for execution. You can specify LGEN as the fourth parameter of the SYSTEM= keyword for the IMSCTRL macro. For information on how to specify LGEN on IMSCTRL, see “IMSCTRL Macro” on page 87. ETO descriptors are not generated during LGEN.

IMS LGEN does not support the ICTL assembler instruction.

**Related Reading:** For an overview of the LGEN configuration, refer to *IMS Version 9: Administration Guide: System*.

## Using DBCTL

The term “online” refers to both a DB/DC and DBCTL environment. The term “DBCTL” refers to a system generated specifically for DBCTL, not to a DB/DC environment performing DBCTL services. Table 15 on page 33 specifies the macros that apply to DBCTL; use this information to identify which macros apply to which IMS configurations.

Any reference to LSO=S refers to the DLISAS in a DBCTL environment. Any reference to MTO also applies to the system operator for DBCTL. References to IMS commands (for example, /START) apply to DBCTL, assuming that the “/” is the DBCTL command recognition character. And, any references to MFS are **not** applicable to a DBCTL environment. All other information applies to DBCTL unless specifically stated otherwise.

The minimum system definition required to generate DBCTL is an ON-LINE type and a DBCTL class; for further information see the SYSTEM parameter of the IMSCTRL macro.



To define a batch nucleus within DBCTL, use BATCH as the type and DBCTL as the class. Use DBCTL class to run BATCH jobs and to process network transactions without IMS TM. This nucleus is used in the batch DL/I environment.

## Using DCCTL

The term “DCCTL” refers to a system generated specifically for DCCTL, which is also an online environment. The system definition process for DCCTL is identical to the process required to define and generate an IMS DB/DC system. You can specify DCCTL as the fourth parameter of the SYSTEM= keyword for the IMSCTRL macro. This causes the appropriate DCCTL environment to be generated. Table 15 on page 33 specifies the macros that apply to DCCTL.

No separate DL/I address space exists in a DCCTL environment.

To define a batch nucleus within DCCTL, use BATCH as the type and DCCTL as the class. This produces a TM batch environment, which can be used to access GSAM or DB2® data. Although the DCCTL environment does not support DL/I databases, DEDBs, or MSDBs, it does support GSAM databases. This support allows applications to issue symbolic checkpoint (CHKP) and extended restart (XRST) calls against the data set to allow data repositioning.

## Using the Macro Table

Table 15 on page 33 shows, in order of complexity, which IMS system definition parameters can be modified by type of system definition. You can override some keywords for some macros by specifying certain keyword parameters on the EXEC statement for the IMS, DBC, or DCC procedure, or by modifying the JCL. Other keywords can also be overridden through IMS operator commands. For further information, see Chapter 8, “Tailoring the IMS System to Your Environment,” on page 327.

Certain macro statements, keywords, and parameters require additional consideration if they are being newly added to or entirely deleted from the system. These are explained in the notes that follow Table 15 on page 33.

To use Table 15 on page 33, find the macro-associated keyword and parameter that you want to change. Then scan across the table to find the first column with an “X” under the system definition types. This identifies the minimum system definition required to change the parameter. If multiple changes are to be made, scan each macro statement on the figure to determine the minimum system definition. When changing operands, make certain the minimum system definition required to update the operand is performed. If you specify a lower level than that indicated in Table 15 on page 33 the results are unpredictable.

The JCL column in the table identifies the EXEC parameter, DFSPBIMS keyword, DFSPBDBC keyword, DFSPBDCC keyword, or IMS, DBC, or DCC operator command to override the operand in the macro.

Keywords and parameters changed during a MODBLKS system definition can be brought online through a series of /MODIFY operator commands, except as noted in Note 1 on page 39.

The CTLBLKS system definition option on the IMSCTRL macro statement can be used **only** to replace the control blocks of an existing nucleus—that is, a nucleus having the same suffix.

Adding new device support features or options might require a NUCLEUS system definition, because additional modules (for example, the VTAM terminal COPY option and module DFSCVEQ0) might need to be bound into the nucleus.

Some of the values can be displayed using the /DISPLAY command shown in table notes. The value in the /DISPLAY column is used to qualify the /DISPLAY command as necessary.

Table 15. Macro Table - Selecting the Appropriate IMS System Definition

Macro	Operand	Value	JCL or Command	/DISPLAY	MODBLKS	CTLBLKS	NUCLEUS	ON-LINE	ALL	MSVERIFY	DB BATCH	TM BATCH	DBCTL	DB/DC	DCCTL	Refer to Table Notes at the end of this table	
APPLCTN	DOPT				X	X	X	X	X				X	X	X		
	PGMTYPE	class	/ASSIGN		X	X	X	X	X					X	X	26	
		All others				X	X	X	X	X				X	X	X	
	RESIDENT				X	X	X	X	X				X	X	X		
	FPATH				X	X	X	X	X					X	X		
	SYSID					X	X	X	X	X					X	X	1
		All Others				X	X	X	X	X				X	X	X	
BUFPOOLS	DMB		DMB=	DMBP		X	X	X	X				X	X		2, 27	
	EPCB		EPCB=	EPCB		X	X	X	X				X	X	X	2, 27	
	FORMAT	size 1	FBP=	MFP		X	X	X	X					X	X	2, 27	
	FRE		FRE=			X	X	X	X					X	X	2	
	PSB		PSB=	PSBP		X	X	X	X				X	X	X	2, 27	
	PSBW		PSBW=	PSBW		X	X	X	X				X	X	X	2, 27	
	SASPSB	size 1	CSAPSB=				X	X	X	X				X	X		2
		size 2	DLIPSB=				X	X	X	X				X	X		2
COMM	COPYLOG		/SMCOPY			X	X	X	X					X	X		
	OPTIONS	NOFMAST FMTMAST					X	X	X					X	X		
		NOMFSTEST MFSTEST						X	X	X					X	X	2
		NOPAGE PAGING					X	X	X					X	X		
		NOMSPEX MSPEXIT					X	X	X	X				X	X	5	
		NOUSEMSG USERMSGS					X	X	X	X				X	X	5	
	All Others					X	X	X	X				X	X			

Table 15. Macro Table - Selecting the Appropriate IMS System Definition (continued)

Macro	Operand	Value	JCL or Command	/DISPLAY	MODBLKS	CTLBLKS	NUCLEUS	ON-LINE	ALL	MSVERIFY	DB BATCH	TM BATCH	DBCTL	DB/DC	DCCTL	Refer to Table Notes at the end of this table
	RECANY					X	X	X	X					X	X	3, 4
	All Others					X	X	X	X					X	X	
CONFIG						X	X	X	X					X	X	
CTLUNIT						X	X	X	X					X	X	
DATABASE	ACCESS		DD		X	X	X	X	X				X	X		1, 6, 7, 38 on page 41, 39 on page 41
			/START		X	X	X	X	X				X	X		1, 28
	RESIDENT				X	X	X	X	X				X	X		
	All Others				X	X	X		X					X		
FPCTRL								X	X				X	X	X	35
	BFALLOC	parm 1	DBFX=			X	X	X	X				X	X		2, 8
		parm 2	DBBF=			X	X	X	X				X	X		2, 8
		parm 3	BSIZ=			X	X	X	X				X	X		2, 8
	OThread		OTHR=			X	X	X	X				X	X		2, 8
	USRHASH		UHASH=			X	X	X	X				X	X		2, 8
IDLIST						X	X	X	X					X	X	
IMSCTF	APNDG							X	X				X	X	X	
	CPLOG					X	X	X	X				X	X	X	
	LOG					X	X	X	X			X	X	X	X	
	PRDR		PRDR=			X	X	X	X				X	X	X	2
	RDS					X	X	X	X				X	X	X	
	SVCNO	type 2						X	X		X	X	X	X	X	9
		type 4						X	X				X	X	X	
IMSCTRL	CMDCHAR		CRC=			X	X	X	X				X			
	DBRC	parm 1	DBRC=						X		X	X	X	X	X	2, 10, 21

Table 15. Macro Table - Selecting the Appropriate IMS System Definition (continued)

Macro	Operand	Value	JCL or Command	/DISPLAY	MODBLKS	CTLBLKS	NUCLEUS	ON-LINE	ALL	MSVERIFY	DB BATCH	TM BATCH	DBCTL	DB/DC	DCCTL	Refer to Table Notes at the end of this table
		parm 2	DBRC=						X		X	X	X	X	X	10, 21
	DBRCNM		DBRCNM=			X	X	X	X				X	X	X	2, 29
	DCLWA					X	X	X	X					X	X	
	DESC					X	X	X	X			X	X	X	X	
	DLINM		DLINM=			X	X	X	X			X	X	X		2, 29
	ETOFEAT							X	X					X	X	
	GSGNAME		GSGNAME=			X		X	X		X	X	X	X	X	36
	HSB							X	X					X	X	12
	IMSID		IMSID=			X	X	X	X			X	X	X	X	2
	IRLM		IRLM=			X	X	X	X			X	X	X		2, 11, 37
	IRLMNM		IRLMNM=			X	X	X	X			X	X	X		2, 11, 37
	MAXCLAS					X	X	X	X			X		X	X	
	MAXREGN		PST=			X	X	X	X				X	X	X	2
	MCS					X	X	X	X			X	X	X	X	
	MSVID					X	X	X	X	X			X	X	X	
	NAMECHK					X	X	X	X			X	X	X	X	
	RSRFEAT							X	X		X	X	X	X	X	
	SYSTEM							X	X		X	X	X	X	X	
	TMINAME		TMINAME=			X		X	X		X	X	X	X	X	36
IMSGEN	MFSDFMT					X	X	X	X					X	X	
	MFSTEST						X	X	X				X	X	X	2
	PSWDSEC					X	X	X	X				X	X	X	
	SECCNT						X	X	X				X	X	X	
	SUFFIX						X	X	X				X	X	X	
	SURVEY								X				X	X		13

Table 15. Macro Table - Selecting the Appropriate IMS System Definition (continued)

Macro	Operand	Value	JCL or Command	/DISPLAY	MODBLKS	CTLBLKS	NUCLEUS	ON-LINE	ALL	MSVERIFY	DB BATCH	TM BATCH	DBCTL	DB/DC	DCCTL	Refer to Table Notes at the end of this table
	SYSMMSG					X	X	X	X					X	X	
	TERMSEC					X	X	X	X					X	X	
	All Others				X	X	X	X	X	X	X	X	X	X	X	
LINE	ADDR		DD			X	X	X	X					X	X	14, 30
	All Others					X	X	X	X					X	X	
LINEGRP						X	X	X	X					X	X	15
MSGQUEUE	BUFFERS	NBR	QBUF=			X	X	X	X					X	X	2
		SIZE				X	X	X	X					X	X	
	All Others					X	X	X	X					X	X	
MSLINK						X	X	X	X					X	X	
MSNAME						X	X	X	X					X	X	
MSPLINK	ADDR		DD			X	X	X	X					X	X	14, 16, 31
	MODETBL		/CHANGE	link# MODE		X	X	X	X					X	X	32
	OPTIONS	asr	/CHANGE			X	X	X	X					X	X	31
	TYPE					X	X	X	X					X	X	16, 17
	All Others					X	X	X	X					X	X	
NAME						X	X	X	X					X	X	18
POOL						X	X	X	X					X	X	19
RTCODE					X	X	X	X	X					X	X	
SECURITY	PASSWD	yes/no	/NRESTART			X	X	X	X					X	X	
	TYPE	AGNEXIT					X	X	X				X	X	X	
		RACFAGN	ISIS=				X	X	X				X	X	X	2
		RACFCOM	RCF=				X	X	X					X	X	2
		RACFTERM	RCF=				X	X	X					X	X	2
		RAS	ISIS=				X	X	X				X	X	X	

Table 15. Macro Table - Selecting the Appropriate IMS System Definition (continued)

Macro	Operand	Value	JCL or Command	/DISPLAY	MODBLKS	CTLBLKS	NUCLEUS	ON-LINE	ALL	MSVERIFY	DB BATCH	TM BATCH	DBCTL	DB/DC	DCCTL	Refer to Table Notes at the end of this table
		RASEXIT					X	X	X				X	X	X	
		RASRACF	ISIS=				X	X	X				X	X	X	
		SIGNEXIT				X	X	X	X					X	X	23
		TRANEXIT				X	X	X	X					X	X	23
	SECCNT					X	X	X	X					X	X	
	SECLVL	SIGNON	/NRESTART			X	X	X	X					X	X	2, 22
		TRANAUTH	/NRESTART			X	X	X	X					X	X	2, 22
	TERMNL	yes/no/force	/NRESTART			X	X	X	X					X	X	
	TRANCMD	yes/no/force	/RESTART			X	X	X	X					X	X	
	All Others					X	X	X	X				X	X	X	
STATION						X	X	X	X					X	X	
SUBPOOL						X	X	X	X					X	X	
TERMINAL	ADDR		DD			X	X	X	X					X	X	15, 20
	KEY		/CHANGE			X	X	X	X					X	X	
	MODETBL		/CHANGE	nodename NAME		X	X	X	X					X	X	33
	OPTIONS	asr	/CHANGE			X	X	X	X					X	X	34
	All Others					X	X	X	X					X	X	
TRANSACT	AOI															
	DCLWA				X	X	X	X	X					X	X	
	EDIT	name				X	X	X						X	X	24, 25
	All Others				X	X	X	X	X					X	X	
	MAXRGN		/CHANGE		X	X	X	X	X					X	X	26
	MSGTYPE	class	/ASSIGN		X	X	X	X	X					X	X	1, 26

Table 15. Macro Table - Selecting the Appropriate IMS System Definition (continued)

Macro	Operand	Value	JCL or Command	/DISPLAY	MODBLKS	CTLBLKS	NUCLEUS	ON-LINE	ALL	MSVERIFY	DB BATCH	TM BATCH	DBCTL	DB/DC	DCCTL	Refer to Table Notes at the end of this table
		All Others			X	X	X	X	X					X	X	
	PARLIM		/ASSIGN		X	X	X	X	X					X	X	1, 26
	PROCLIM	count	/ASSIGN		X	X	X	X	X					X	X	1, 26
		All Others			X	X	X	X	X					X	X	
	PRTY		/ASSIGN		X	X	X	X	X					X	X	1, 26
	SEGNO		/ASSIGN		X	X	X	X	X					X	X	1, 26
	SEGSIZE		/ASSIGN		X	X	X	X	X					X	X	1, 26
	SYSID				X	X	X	X	X					X	X	1
	All Others				X	X	X	X	X					X	X	
TYPE	OPTIONS		SIGNON=			X	X	X	X					X	X	4, 15
VTAMPOOL						X	X	X	X					X	X	



**Notes to Table 15 on page 33:**

1. Changes to existing transactions cannot be introduced via a MODBLKS system definition if those attributes are changeable by online commands.
2. Refer to IMS, DBBATCH, DBC, DCC, and DLIBATCH procedures in Chapter 4, "Procedures," on page 191.
3. This keyword can be changed with any type of system definition except MODBLKS.
4. Adding or deleting VTAM support requires the minimum of an ON-LINE system definition.
5. Adding a user-supplied exit routine, user message table, or the shared printer message router exit routine requires the minimum of a CTLBLKS system definition. The exceptions to this requirement are the ETO, message greeting, and command security exit routines. To remove the use of these exit routines requires a minimum of NUCLEUS system definition. These exit routines are then loaded or deleted during initialization if they are in IMS.SDFSRESL.
6. The ACCESS attribute is used in data sharing and can be dynamically changed with the /START command.
7. ACCESS can be forced to exclusive by specifying DISP=OLD in the database DD statement or in the DFSMDA definition.
8. Adding or deleting Fast Path support for the FPCTRL statement requires a minimum of an ON-LINE system definition. The FPBUF parameter of the TERMINAL macro is not required to enable Fast Path for a specific terminal. Fast Path is automatically enabled for all Fast Path-capable terminals.
9. Although the value for the Type 2 SVC number is changed in the necessary control blocks for all system definition types, only the ON-LINE and ALL types of system definition bind the SVC module into IMS.SDFSRESL with the proper name required for subsequent bind into the operating system libraries.
10. These parameters can only be **excluded** using JCL. If their inclusion in your system is required, they must be included using system definition parameters.
11. For an online system, modifying the IRLM= or IRLMNM= parameters requires a minimum of a CTLBLKS type system definition. A batch system requires a minimum of either a BATCH or an ALL system definition.
12. Generating a system for use in an IMS backup configuration requires a minimum of an ON-LINE system definition.
13. Adding Database Surveyor support to an IMS system initially generated without it can be done in two ways. If done via system definition, either a BATCH system definition (for batch) or an ALL system definition (for both batch and online) must be specified to include Surveyor. Alternatively, in either a batch or online system, the following JCL can be run:

```
//LINKSUR JOB
//          EXEC PGM=IEWL,REGION=128K,
//          PARM='NCAL,LET,XREF'
//SYSPRINT DD SYSOUT=A
//LOAD     DD DSNAME=IMS.ADFSLOAD,DISP=SHR
//SYSLOAD  DD DSNAME=IMS.SDFSRESL,DISP=SHR
//SYSUT1   DD SPACE=(CYL,(1,1)),UNIT=SYSDA
//SYSLIN   DD *
//          INCLUDE LOAD(DFSPRSUR)
//          INCLUDE LOAD(DFSLI000)
//          NAME DFSPRSUR(R)          DATABASE SURVEYOR
//          /*
```

14. Change the DD statement.
15. Adding or deleting support for a specific non-ETO terminal type requires a minimum of a NUCLEUS system definition. A minimum of a NUCLEUS system

definition is required if the SIZE= or FEAT= parameter is specified for the first time in an existing system or if the values of these parameters are changed. You can use any type of system definition except MODBLKS to add, delete, or change this macro statement.

16. Adding MSC support for the first time or deleting MSC support requires a minimum of an ON-LINE system definition. A physical link change requires a DD statement change.
17. Adding or deleting the BSC or channel-to-channel support requires a minimum of an ON-LINE system definition. Adding MSC VTAM or MTM support requires a minimum of an ON-LINE system definition. Deleting MSC VTAM or MTM support requires a minimum of a NUCLEUS system definition.
18. Removing remote LTERM support from an MSC system requires a minimum of a NUCLEUS system definition.
19. Adding or deleting switched terminal support requires a minimum of a NUCLEUS system definition.
20. The address of 3270 local terminals can be changed on the DD statements. All other terminals require a minimum of a CTLBLKS system definition.
21. When changing this parameter to FORCE, or from FORCE to another option, either a BATCH or an ALL system definition is required for the change to take effect in a batch environment.
22. JCL can be used to override SIGNON and TRANAUTH only if SIGNON is required or TRANSACT authorization is specified. The signon and transaction authorization exit routines can still be used if you sign on.
23. To remove exit routine support requires a minimum of a NUCLEUS system definition defined for static terminals.
24. MODBLKS system definition cannot be used to add a transaction that is also adding a new user edit routine. MODBLKS system definition cannot be used to change the order of the transactions if user edit routines are specified. The minimum system definition for either of these occurrences is a CTLBLKS system definition.
25. MODBLKS system definition cannot be used to delete a transaction that has a user edit routine if this is the first or only transaction that is using this edit routine. MODBLKS system definition cannot be used to change the order of the transactions if user edit routines are specified. The minimum system definition for either of these occurrences is a CTLBLKS system definition.
26. The command is /DISPLAY TRANSACTION.
27. The command is /DISPLAY POOL xxxx, where xxxx is the indicated parameter.
28. The command is /DISPLAY DATABASE.
29. The command is /DISPLAY ACTIVE.
30. The command is /DISPLAY LINE.
31. The command is /DISPLAY LINK.
32. The command is /DISPLAY LINK xxxx, where xxxx is the indicated parameter.
33. The command is /DISPLAY NODE.
34. The command is /DISPLAY MODE xxxx, where xxxx is the indicated parameter.
35. If Fast Path is to be included in a DCCTL environment, specify the FPCTRL macro without any parameters.
36. For batch systems, specify the parameter in the JCL; for online systems, specify the parameter in the DFSRSRxx member.
37. A BATCH or ALL system definition is required for the change to take effect in a batch environment.

- | 38. A database that is defined in an ALL or MODBLKS system definition cannot be  
| converted into a HALDB partition without a cold start of IMS. A cold start is  
| required even if the database is deleted online. After the cold start, the  
| database must be redefined as a HALDB partition.
- | 39. A database that is defined as a HALDB partition to DBRC and IMS cannot be  
| redefined in an ALL or MODBLKS system definition without a cold start of IMS.  
| A cold start is required even if the database is deleted online.

---

## Managing System Definition

System definition is a two-stage process. Stage 1 checks your input specifications and generates a series of z/OS job steps for stage 2. Stage 2 builds IMS system libraries, execution procedures, and the IMS control program.

In an XRF environment, the system definition of the active and alternate IMS subsystems must be identical, although two separate system definitions can occur.

**Related Reading:** See “Designing a System with Extended Recovery Facility” in *IMS Version 9: Administration Guide: System* for information on defining an XRF system.

## Sequencing Stage 1 Input

During the initialization for startup of the IMS online system, a control program, which has been customized to your requirements, is loaded. The input data that drives this process is a set of macro source statements. Each statement is coded with its own parameters. The composite of all the macro statements is termed the *stage 1 input*.

Four sets or groupings of macro statements make up the content of the stage 1 input. (The IMSGEN macro contributes to the system configuration definition.) Within each set, individual macros specify data that is specific to a required function or to a part of the total physical online configuration.

You can think of these groupings of macros as a type of hierarchical structure, as shown in Figure 9 on page 42. The group of required system configuration macros is shown as a root segment.

Hierarchy of Stage 1 System Definition Macros

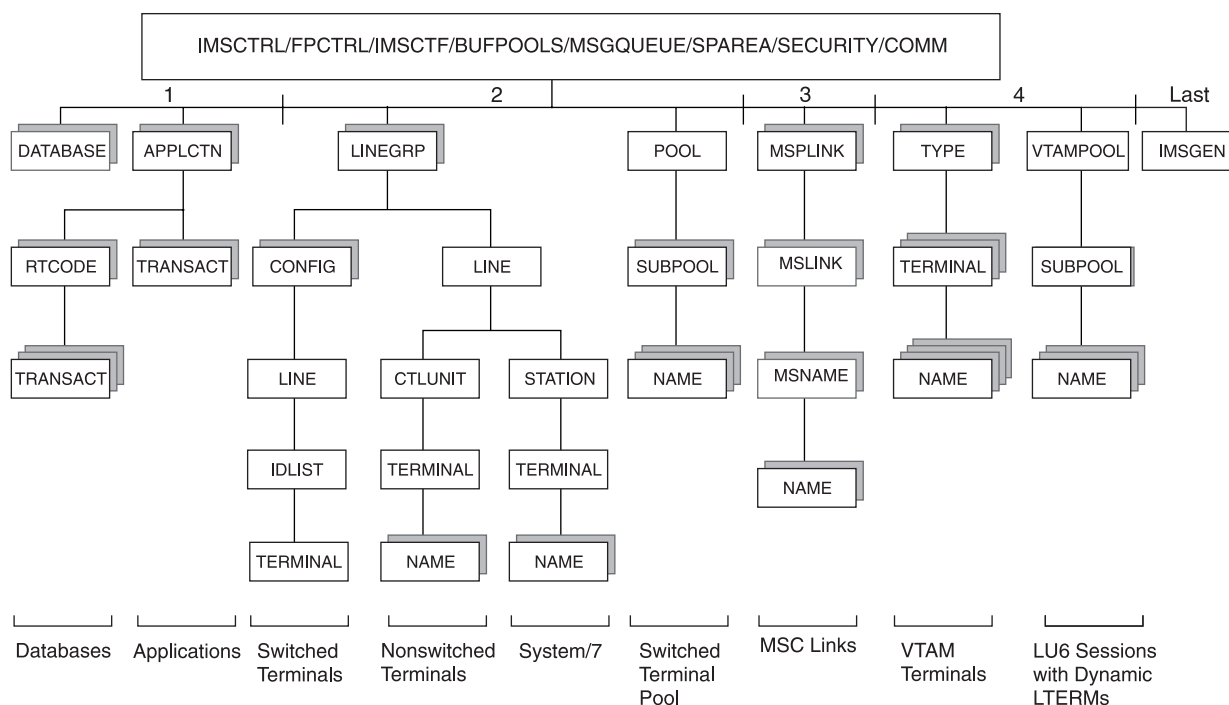


Figure 9. Hierarchy of Stage 1 System Definition Macros

**Restriction:** The IMSGEN macro must be the last macro defined.

## LGEN System Definition Changes

Two steps are required to define a large system generation:

1. In the IMSCTRL macro statement, specify LGEN as the fourth subparameter of the SYSTEM keyword. Define all other macro statements for the system definition as you normally do.
2. Allocate and catalog two additional data sets needed for the Stage 2 processing in the LGEN configuration. The default names for these data sets are IMS.LGENIN and IMS.LGENOUT. You can use the NODE keyword of the IMSGEN macro to override the default names. To allow sufficient storage, allocate ten primary and five secondary cylinders for each data set, and specify a PDS size of 40 blocks.

## LGEN Stage 1 Processing

Stage 1 processing for LGEN system generation differs from a standard system definition. You must execute the preprocessor to assemble an LGEN generation. A stand-alone assembly of an LGEN generation causes an error. Stage 1 source statements are used as input and can include copy statements. However, the stage 1 input source statements cannot include inline copies of IMS macros, inline user macros, or user macro calls.

In turn, the preprocessor checks the resource name. The preprocessor uses storage above the 16 MB line and calls the assembler multiple times, called "cycles", to assemble all the input. This ensures that the stage 1 execution requires only 4 MB of private storage. You receive two reports from stage 1 processing: a summary of the return codes for each assembly and a summary of error messages.

The result of the assembly cycles is:

- A job stream that creates members of a PDS for the IMS control blocks
- The job steps that are dictated by the system definition type

The PDS members are held in the LGENIN data set and are used as input to the Sort/Split utility. The Sort/Split utility sorts all the PDS members for a given resource type and then splits them into parts that are to be assembled in a 4 MB region. Each part becomes a member of a PDS stored in the LGENOUT data set. The Sort/Split utility supplies a report about its processing.

The Sort/Split utility processes each resource type individually, including both standard and customized resources. Customized resources, referred to as “user-defined resources”, are normally created by modifying stage 1 macros. A control record is required for each user-defined resource in the Resource Information File member. This record tells the Sort/Split utility to process the resource.

**Related Reading:** For more information on user-defined resources, see *IMS Version 9: Customization Guide*.

## LGEN Stage 2 Processing

LGEN stage 1 processing creates altered job steps for stage 2. However, the differences in stage 2 processing (between LGEN and standard) are not obvious because LGEN process stage 2 inputs into assemble and binder jobs that are similar to the standard stage 2 job steps.

---

## Verifying System Definition

To verify system definition, you can use the system definition preprocessor to scan the stage 1 input. The preprocessor checks for duplicate names among the names you have defined and ensures that they are of the correct length and format. Assigned names are checked across resource types, too, so that transaction codes, LTERMs, and IMS subsystem names (used for multiple systems coupling) do not contain duplications. The preprocessor helps maintain the integrity of the stage 1 input stream.

## Preprocessor

The system definition preprocessor is optional unless the LGEN configuration has been specified in the IMSCTRL SYSTEM keyword. Use the LGEN configuration when more than 4 MB of private storage are needed in stage 1 or in any stage 2 assembly job step.

If the IMSCTRL NAMECHK=NO option is selected for IMS system definition, run the system definition preprocessor.

In the LGEN configuration, the IMSCTRL NAMECHK=NO option is ignored and need not be specified. Under LGEN, no sorting is done in the stage 1 or stage 2 assemblies. Sorting is done in the preprocessor and by the Sort/Split utility.

Stage 1 source statements are used as input and can include copy statements. However, they cannot include inline copies of IMS macros from the stage 1 input, inline user macros, user macro calls, or conditional assembler statements.

When running LGEN, do not include assembler listing control instructions (such as PRINT OFF) in the stage 1 source statements. If you do so, you may have unpredictable results, because LGEN processes the output from the assembler listing.

The preprocessor locates, reads, and processes copy statements by using the copy members as input. The order of search, which conforms to z/OS standards, depends upon the order in which you have concatenated the libraries. Copy members are retrieved from the data sets specified by SYSLIB DD statements. The preprocessor scans the input for the following IMS macros:

APPLCTN	NAME
DATABASE	RTCODE
MSLINK	SUBPOOL
MSNAME	TERMINAL
MSPLINK	TRANSACT

The preprocessor does not recognize keywords that are split across continuation statements. The related parameters associated with these keywords generate error messages from the preprocessor.

After scanning for macros, the preprocessor builds resource name tables for each resource name type. It then performs the following verification steps:

- Within each resource type, the preprocessor verifies that no resources of that type have duplicate names. The following types of resource names are verified:
  - DBD names
  - PSB names
  - VTAM node names
  - MS link names
  - Logical terminal names
  - Transaction codes
  - Routing codes
  - Subpool names
  - MSLINK partner IDs
  - MSPLINK physical link names
  - Remote system VTAM node names
- These resource names are checked to ensure that they are of the appropriate length and are alphanumeric.
- The names specified for transactions, logical terminals, and multiple systems are cross-checked to ensure that no duplicate names exist across these three resource types.

When the preprocessor executes with LGEN specified, the following additional processing occurs:

- Stage 1 source statements are saved in storage above the 16 MB line according to resource type.
- The specifications for the system identification (SYSID) keyword on APPLCTN, TRANSACT, and MSNAME stage 1 source statements are checked for syntax errors. They are also cross-checked to ensure that no conflicts exist between the **local** system identification keywords and the **remote** system identification keywords.

- If no errors are detected by the preprocessor, the assembler is called multiple times to process the stage 1 source statements that were saved. Input to each assembly consists of all system macro statements and up to 10000 of either DB, MSC, or VTAM resources. All BTAM resources are processed during a single execution of the assembler.

The output of the preprocessor is a listing of the input records, a listing of diagnostic and error messages, and a return code.

If an LGEN configuration is active, the output includes the following:

- Two summary reports that are printed before any other preprocessor output. The first report contains the return code for each assembly cycle and the total number of resources processed by the preprocessor. The second report lists each error message in an assembly cycle and the five records that preceded the error message.
- Information from the listing of each assembly is appended to the preprocessor listing to form a consolidated listing data set. A list of the cycle macros is included when any assembly fails with a return code greater than 4. This list aids problem determination.
- The object data set from each assembly is appended to the object data sets of all previous assemblies to form a consolidated object data set. Execution of the job stream in this data set is functionally equivalent to stage 2 processing of a standard system definition for fewer resources.

You can optionally write two exit routines, DFSPRE60 and DFSPRE70, to perform additional processing.

**Related Reading:** Information about the register and coding requirements for the DFSPRE60 and DFSPRE70 exit routines can be found in *IMS Version 9: Customization Guide*.

## Executing the Preprocessor

To execute the preprocessor, you must provide JCL as shown in Figure 10 on page 46.

```

//          JOB
//          EXEC PGM=DFSPRE00,REGION=32M,PARM=' xxx' 1
//STEPLIB DD DSN=IMS.SDFSRESL,DISP=SHR 2
//          DD DSN=USER.EXITLIB,DISP=SHR 3
//SYSLIB  DD DSN=USER.MACLIB,DISP=SHR 4
//          DD DSN=IMS.SDFSMAC,DISP=SHR 4
//SYSIN80 DD DSN=&&SYS80, 5
// UNIT=SYSDA,SPACE=(CYL,(5,5)), 5
// DCB=(BLKSIZE=11440,LRECL=80,RECFM=FB,DSORG=PS) 5
//SYSUT1  DD DSN=&SYSUT1, 5
// UNIT=SYSDA,SPACE=(CYL,(10,5)) 5
//SYSLIN  DD DSN=&&SYSLN, 5
// UNIT=SYSDA,SPACE=(CYL,(1,1)), 5
// DCB=(BLKSIZE=11440,LRECL=80,RECFM=FB,DSORG=PS) 5
//SYSPRT80 DD DSN=&&SYSPRT, 5
// UNIT=SYSDA,SPACE=(CYL,(5,5)), 5
// DCB=(BLKSIZE=3146,LRECL=121,RECFM=FBM,DSORG=PS) 5
//SYSSUMPR DD SYSOUT=A 6
//SYSCYLPR DD SYSOUT=A 6
//SYSPRINT DD SYSOUT=A
//SYSCOBJ DD DSN=IMS.SYSCOBJ,DISP=(NEW,KEEP), 7
// UNIT=SYSDA,SPACE=(CYL,(5,5)), 7
// DCB=(BLKSIZE=11440,LRECL=80,RECFM=FB,DSORG=PS) 7
//SYSIN   DD * 8

```

Figure 10. JCL to Execute the Preprocessor

#### Notes:

1. The region size indicated is the size required to execute the preprocessor on an OS/390 system. The LGEN subparameter is specified in the SYSTEM= keyword on the IMSCTRL source statement. Guidelines for determining the region sizes without LGEN are in Estimating Storage Requirements for the Preprocessor. The PARM field is specified only if the default exit routine indicators should be overridden. See the following section, "Defining Preprocessor Parameters" on page 47, for information on use of the PARM field.
2. This DD statement should specify the library containing the preprocessor version that you want to execute. Different versions can exist because of SMP maintenance.
3. This concatenated DD statement is required only if exit routines are requested.
4. The SYSLIB DD statement should point to a library that contains the copy members, but with an LGEN definition, the statement should point to desired system definition macro instruction libraries. This DD statement is required if stage 1 source COPY statements or an LGEN configuration exists.
5. Only the LGEN configuration requires these DD statements.
6. Only the LGEN configuration requires these DD statements. These DD statements must appear before the SYSPRINT DD statement to allow the summary reports to precede the consolidated assembly listing.
7. Only the LGEN configuration requires this data set; it contains the consolidated object module from all the LGEN assembler cycles. The input/output unit assigned to this data set can be a card punch or an intermediate storage device capable of sequential access.
8. You can also use the statement:

```
//SYSIN DD DSN=....,DISP=SHR
```



## Defining Preprocessor Parameters

The PARM field on the EXEC statement is specified as follows:

```
PARM='a,b,(asmopt1,asmopt2,...,asmoptn)'
```

The parameters specified are positional and are specified as follows:

- a** Is a value of 'Y' or 'N' and indicates whether (Y) or not (N) exit routine DFSPRE60 is to be used during this invocation of the preprocessor. If DFSPRE60 is to be used, it must reside on the libraries pointed to by the STEPLIB DD statement. The default is N.
- b** Is a value of 'Y' or 'N' and indicates whether (Y) or not (N) exit routine DFSPRE70 is to be used during this invocation of the preprocessor. If DFSPRE70 is to be used, it must reside on the libraries pointed to by the STEPLIB DD statement. The default is N.

### asmopt1, ..., asmoptn

Are any assembler options you want for stage 1 assemblies in the LGEN configuration. Enclose the options in parentheses, and use the comma as a delimiter when using more than one option. The preprocessor forces the LIST and OBJECT options.

**Related Reading:** For more information on the LGEN configuration, refer to *IMS Version 9: Administration Guide: System*.

## Estimating Storage Requirements for the Preprocessor

For both the standard and LGEN system definitions, the preprocessor uses the default values to reserve an initial amount of storage for each resource name table (RNT). If this storage is insufficient, the preprocessor dynamically expands the table until all extended private storage is exhausted. Resource names not added to the appropriate RNT are ignored and excluded from further processing.

Additional space is required if the number of errors for any given resource name exceeds 50, or if the number of nested copy statements exceeds 50. For most cases, this additional storage is accounted for in the base system storage number.

The storage size is the minimum region size that can be specified on preprocessor invocation JCL.

The preprocessor also requires one megabyte of private storage below the 16 MB line.

### Standard system definition

The storage estimate is for extended private storage. The storage requirement is the sum of:

1. 1000 KB for the base system
2. Space needed for exit routines DFSPRE60 and DFSPRE70, if they are loaded
3. Table space in excess of the default. For each of the following resource types that exceeds 5120, determine the amount of storage from Table 16 on page 48.

APPLCTN

DATABASE

NAME

RTCODE

SUBPOOL

TERMINAL (VTAM)  
 TRANSACT

Table 16. Resource Name Table Storage

Resource Range	Storage Amount n K bytes
5120 to 10240	80
10240 to 20480	240
20480 to 41060	560
41060 to 82120	1200
82120 to 164240	2480
164240 to 328480	5040
328480 to 656960	10160
656960 to 1313920	20400

### LGEM System Generation

If the total number of resources is less than 200000, allow 32 MB for extended private storage.

When the total number of resources exceeds 200000 resources, determine the amount of extended private storage that is required.

The sum of the following items, numbered 1 and 2, is the total extended private storage (in kilobytes) required to run the preprocessor with an LGEM system generation. The preprocessor also requires 4 MB of private storage below the 16-MB line.

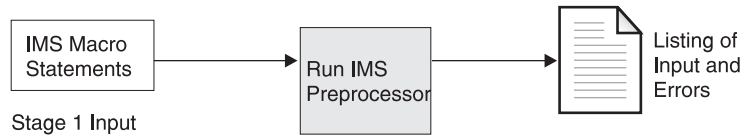
LGEM limits the amount of private storage required to 4 MB for DB, MSC, and VTAM resources. All BTAM resources are assembled in a single assembly cycle. A large number of BTAM resources can require more than 4 MB of private storage for the stage 1 and stage 2 assemblies.

1. Storage for system definition input is the sum of:
  - 3000 for the base system
  - ((Number of system definition input records/8192)x 584)
  - 584 if any BTAM resources are defined
  - 584 if any MSC resources are defined
  - 584 if any VTAM resources are defined
2. Table space in excess of default. For each of the following resource types that exceeds 5120, determine the amount of storage from Table 16.
  - APPLCTN (total)
  - APPLCTN (remote)
  - DATABASE
  - NAME
  - RTCODE
  - SUBPOOL
  - TERMINAL (VTAM)
  - TRANSACT (total)
  - TRANSACT (remote)

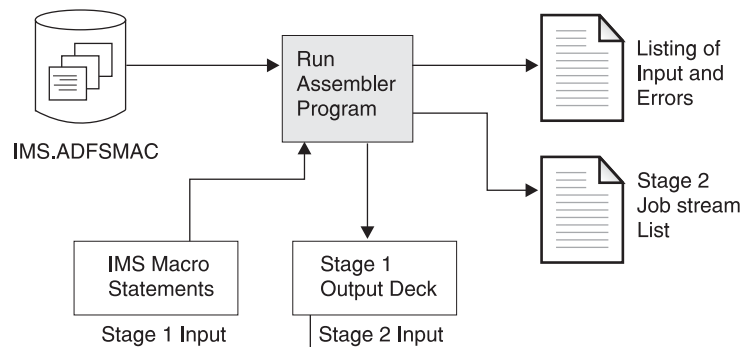
## Running System Definition Stage 1

After coding your macro statement input (and optionally running the preprocessor), you must now run your IMS system definition stage 1. The preprocessor runs stage 1 in an LGEN configuration. The stage 1 output is the input to system definition stage 2. The two stages of the system definition process are illustrated in Figure 11.

### IMS System Definition: Preprocessor



### IMS System Definition: Stage 1



### IMS System Definition: Stage 2

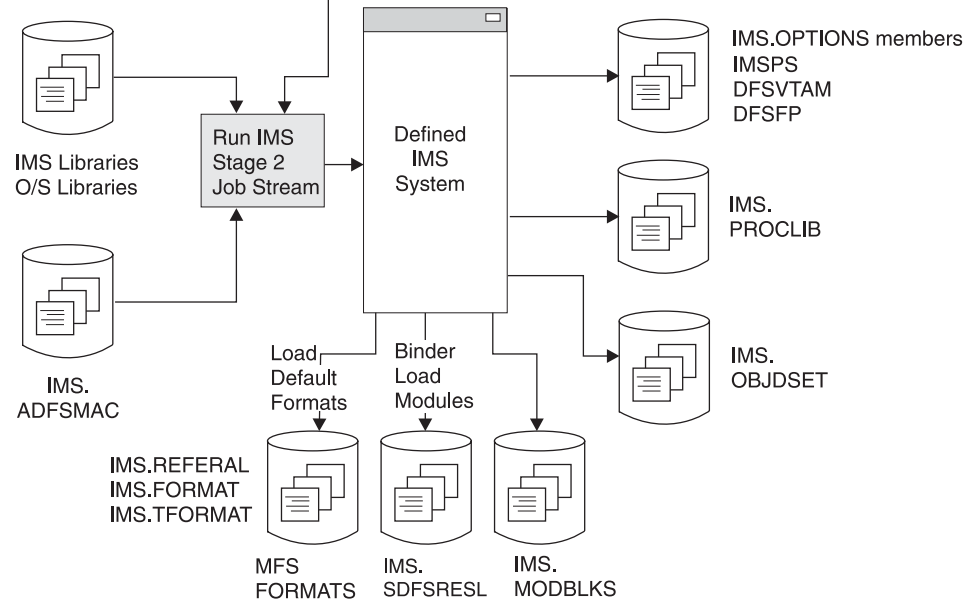


Figure 11. Summary of the Two Stages of System Definition Processing

## Entering Stage 1 Input Statements

Stage 1 of IMS system definition consists of running the assembler with the IMS macro statements that you have prepared as input. The macro sets must be placed

in the order described in “Sequencing Stage 1 Input” on page 41. If more than 5000 objects are defined, sufficient storage must be allocated for the system definition process.

## Stage 1 and Extended Terminal Option

If you are using the Extended Terminal Option (ETO) of IMS, you can run stage 1 in three different ways.

- You can use the existing stage 1. In this case, the default ETOFEAT is (,NO) on the IMSCTRL macro, and the output from stage 1 is the same as described in Figure 11 on page 49.
- You can code ETOFEAT=(,ONLY) on the IMSCTRL macro to produce ETO descriptors only.
- You can code ETOFEAT=(,ALL) to produce ETO descriptors along with a normal system definition.

For more information on ETO, see “Enabling ETO Support for ACF/VTAM Terminals” on page 450.

## Stage 1 Output

Output that results from the successful execution of the stage 1 input includes:

- The z/OS job stream for stage 2.
- An output listing that also includes a listing of the input statements you submitted and comments about the system you defined.

### Printed Output

The printed stage 1 output lists the input statements that you prepared, and the output. Input errors or inconsistencies are noted on the listing. Included are explanations of tasks that must be accomplished before stage 2 of IMS system definition can be executed.

**Related Reading:** If errors are encountered during IMS system definition stage 1, error messages that appear on the output listing are explained in *IMS Version 9: Messages and Codes, Volume 1*; *IMS Version 9: Messages and Codes, Volume 2*; or in the appropriate assembler language messages manual.

One coding error that frequently occurs is beginning a continued statement in column 17 or later. The assembler does not flag this as an error. Rather, the improperly continued statement is treated as a comment. IMS therefore assumes the appropriate default values for any operands that were coded on the continuation statement and ignores any values you might have specified.

### Binder Order Statements

The ORDER statements produced by stage 1 must be used when binding the IMS nucleus.

### Stage 2 Input

The stage 1 output is the input to IMS system definition stage 2. If possible, you should list these statements again for easier use and become familiar with them before performing stage 2.

Some or all of the job steps are described below, depending on the type of IMS system definition specified in the IMSCTRL macro statement.

- Creation of IMS.OPTIONS containing IMSPS, DFSVTAM, DFSFP, and DFSIRLM.

- The IMSPS member is used during the APAR maintenance process.
- The DFSVTAM member is used in the stage 2 assembly of the SYSGEN-created module, DFSCCLLOx, where x is the SYSGEN suffix. It provides information as to whether VTAM or BTAM terminals were defined in stage 1.
- The DFSFP member is used by conditional assemblies during stage 2 processing. It provides information as to whether Fast Path was defined in stage 1.
- The DFSIRLM member is used by conditional assemblies during stage 2 processing. It provides information as to whether the IRLM component is defined in stage 1.
- Creation of IMS.PROCLIB.
- Assemblies of online control blocks into IMS.OBJDSET.  
In some cases, the region size generated for the jobs that assemble the IMS control blocks cannot be adequate.  
If a region larger than 4 MB seems necessary, running the stage 1 with an LGEN configuration ensures that stage 2 job steps require no more than 4 MB.
- Binding of reentrant, reusable, and default attribute load modules and control block modules into IMS.SDFSRESL and IMS.MOBLKS.
- Program executions to load the Message Format Service (MFS) default formats into IMS.REFERAL, IMS.FORMAT, and IMS.TFORMAT.

The complete output from stage 1 must be entered into the OS/390 job stream in the order that it was produced. This initiates stage 2 of your IMS system definition.

---

## Running System Definition Stage 2

You must put the stage 1 output job stream in your z/OS system's input job stream to initiate stage 2 of the IMS system definition. If stage 2 processing is to be multiprogrammed, the job step that creates IMS.OPTIONS must complete before any assemblies are run. Assemblies must complete before binding is performed and the nucleus bind job must not run until other binding is complete. If stage 2 processing is to be multiprogrammed, then the assemblies must be run **sequentially** rather than in parallel. Successful execution of stage 2 of system definition completes this IMS installation step. All resources, except communication lines and message processing regions, are generated in a "started" mode; a /START command is not required to make the resource operational.

## Stage 2 Output

The JCL from stage 1, which is used as input to stage 2, shows that SYS1.SDFS MAC and SYS1.MODGEN are concatenated to any or all of IMS.OPTIONS, or IMS.SDFS MAC. As in all concatenated libraries, the block sizes of the first library must be at least as large as the largest block size specified for the libraries concatenated to it.

The assembly and bind listings produced during stage 2 execution indicate whether execution was successful. All of these job steps should complete with a 0 return code.

If condition code 4 is set in the job that creates MFS default formats, it should be ignored.

## Maintaining Stage 1 Output with SMP/E

To ensure that your IMS system is installable and maintainable via SMP/E, you must run the SMP/E JCLIN against your stage 1 SYSPUNCH output.

When you run the SMP/E JCLIN against your stage 1 SYSPUNCH output, the SMP/E target zone (TLIB), or data sets accessed to process the functions provided by a product, will be updated to accurately represent the running system. That is, the make up of load modules will be recorded, the correct versions of the assem entries will be recorded, and the correct target library relationships will be built for libraries involved in IEBCOPY (for example, the pre-V7 IMS.MACLIB). The IEBCOPY uses SYSIN DDs to identify DD names for the copy from and copy to DD names; then, the IEBCOPY builds the MOD, MAC, and SRC entries to relate the libraries.

SMP/E JCLIN must be run against the stage 1 SYSPUNCH output of all types of system definition to ensure that your SMP/E control data set remains current. If the SMP/E JCLIN is not run you may encounter errors the next time you try to process service using SMP/E. Additionally, you may not, initially, receive error messages but the system may fail to come up or abend later on in usage. The most common result is an ABENDU0168 during startup because the number of terminals in the SCD does not match the number of terminal blocks.

However, note that the SMP/E JCLIN JCL will vary depending on the SYSGEN configuration used:

- Normal SMP/E JCLIN can be used if standard SYSGEN is used.
- Steps involving DFSIVG10 must be used if LGEN is used.

No matter which SYSGEN configuration you have a return code of 0 is expected from the actual JCLIN step.

## Large System Generation SMP/E JCLIN

LGEN imposes some changes in the way SMP/E JCLIN is performed:

- JCLIN must be run either after Stage 2 or after the LGEN utility steps (at the beginning of the Stage 2) are run.
- The LGEN Stage 2 must be converted to a non-LGEN format before JCLIN is run. Program DFSIVG10 is provided for this purpose.

## Sample JCLIN Job

The following procedure is an example of the JCLIN job that you should run. This is an example using SMP/E. If you are using SMP/4, you must also change the procedure name for the JCLIN processing step. Use the job stream built from this example for both LGEN and non-LGEN jobs.

```
//      JOB
//*
//*****
//* NOTE:  FOR NON-LGEN SYSTEM DEFINITIONS, SMP JCLIN PROCESSING MAY
//*       BE RUN EITHER BEFORE OR AFTER STAGE2.
//*
//* NOTE:  FOR LGEN SYSTEM DEFINITIONS, SMP JCLIN PROCESSING MAY ONLY
//*       BE RUN EITHER:
//*
//*       1 - AFTER STAGE2 PROCESSING IS COMPLETE.
//*       2 - AFTER THE LGEN STEPS AT THE BEGINNING OF THE STAGE2
//*          JOB STREAM HAVE BEEN RUN.
```

```

//*****
//*
//* SCRATCH THE TEMPORARY DATA SET USED BY THE STAGE2 POST PROCESSOR
//* PROGRAM.
//*
//SCRATCH EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DELETE IMS.STG2OUT -
    NONVSAM SCRATCH
//*
//* ALLOCATE THE TEMPORARY DATA SET USED BY THE STAGE2 POST PROCESSOR
//* PROGRAM.
//*
//ALLOCATE EXEC PGM=IDCAMS,DYNAMNBR=200
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  ALLOCATE -
    DSNAME('IMS.STG2OUT') -
    FILE(STG2OUT) -
    RECFM(F,B) -
    LRECL(80) -
    BLKSIZE(6160) -
    DSORG(PS) -
    NEW CATALOG -
    SPACE(1,5) CYL -
    VOL(IVPUTL) -
    UNIT(3390)
//*
//* POST PROCESS THE STAGE2 JOB STREAM PRIOR TO SMP JCLIN.
//*
//* FOR NON-LGEN SYSTEM DEFINITONS: THE STAGE2 JOB STREAM IS COPIED,
//* UNCHANGED TO THE TEMPORARY DATASET.
//*
//* FOR LGEN SYSTEM DEFINITONS: AS THE STAGE2 JOB STREAM IS COPIED,
//* TO THE TEMPORARY DATASET, THE MEMBERS OF THE LGENOUT DATA SET ARE
//* PLACED IN-LINE IN THE STAGE2 JOB STREAM PRODUCING A STAGE2 JOB
//* STREAM THAT LOOKS LIKE A NON-LGEN STAGE2.
//*
//POSTPROC EXEC PGM=DFSIVG10
//STEPLIB DD DISP=SHR,DSN=IMS.SDFSRESL
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//STG2IN DD DISP=SHR,DSN=IMS.STG2IN
//STG2OUT DD DISP=OLD,DSN=IMS.STG2OUT
//LGENPDS DD DISP=SHR,DSN=IMS.LEGNOUT
//* PERFORM SMP JCLIN PROCESSING
//JCLIN EXEC PGM=GIMSMP,TIME=(600),DYNAMNBR=250,
// COND=(4,LT,POSTPROC),
// PARM='DATE=U,CSI=IMS.GLBLZONE.CSI'
//*
//SMPCNTL DD *
  SET BDY(TARGET).
  JCLIN ASM(PGM=ASMA90) JCLINREPORT.
//SMPJCLIN DD DISP=OLD,DSN=IMS.STG2OUT

```

The data set names, DASD volume name, and DASD volume device might need to be changed.





---

## Chapter 3. Macros

Use macro statements (such as, APPLCTN or TERMINAL) to generate a sequence of assembler statements from a single source statement. IMS provides a set of macro statements for system definition. For example, to describe program resource requirements for your application programs, you would use the APPLCTN macro. To define characteristics of VTAM nodes and non-VTAM communication, use the TERMINAL macro.

### **In this Chapter:**

- “Guidelines for IMS Macros” on page 59
- “APPLCTN Macro” on page 62
- “BUFPOOLS Macro” on page 67
- “COMM Macro” on page 70
- “CONFIG Macro” on page 78
- “CTLUNIT Macro” on page 79
- “DATABASE Macro” on page 80
- “FPCTRL Macro” on page 82
- “IDLIST Macro” on page 84
- “IMSCTF Macro” on page 85
- “IMSCTRL Macro” on page 87
- “IMSGEN Macro” on page 98
- “LINE Macro” on page 111
- “LINEGRP Macro” on page 114
- “MSGQUEUE Macro” on page 118
- “MSLINK Macro” on page 123
- “MSNAME Macro” on page 124
- “MSPLINK Macro” on page 125
- “NAME Macro” on page 129
- “POOL Macro” on page 134
- “RTCODE Macro” on page 134
- “SECURITY Macro” on page 135
- “SPAREA Macro” on page 140
- “STATION Macro” on page 140
- “SUBPOOL Macro” on page 141
- “TERMINAL Macro” on page 143
- “TRANSACT Macro” on page 174
- “TYPE Macro” on page 188
- “VTAMPOOL Macro” on page 190

This chapter contains the details for all the IMS macros. Table 18 on page 58 contains a list of these macros. Table 15 on page 33 details which macros and macro keywords can be modified by which type of system definition. Examples of macro statement sets are shown in Appendix A, “IMS System Definition (Step 3) Examples,” on page 485.

For information on how to interpret syntax diagrams, see “How to Read Syntax Diagrams” on page xvii in the preface of this book.

Table 17. System Definition Macros and Their Uses

Type of Macro Statement	Macro Name	Resources Each Macro Defines
<b>System Configuration Macros</b>	BUFPOOLS Macro FPCTRL Macro IMSCTF Macro IMSCTRL Macro MSGEN Macro MSGQUEUE Macro SECURITY Macro	Buffer pool sizes Fast Path options System control integrity factors Library naming, JCL options, and ETO feature z/OS system and control options Message queue device and message sizes Security options
<b>Database and Application Macros</b>	APPLCTN Macro DATABASE Macro RTCODE Macro TRANSACT Macro	<ul style="list-style-type: none"> <li>• Program characteristics and PSB names DBD names</li> <li>• Fast Path message routing</li> <li>• Transaction processing options</li> </ul>

Table 17. System Definition Macros and Their Uses (continued)

Type of Macro Statement	Macro Name	Resources Each Macro Defines
<b>Data Communication Macros</b> BTAM, BSAM, GAM, ARAM	COMM Macro CTLUNIT Macro LINE Macro LINEGRP Macro NAME Macro STATION Macro TERMINAL Macro	<ul style="list-style-type: none"> <li>• General communication options</li> <li>• 2848, 2972, 3271 control</li> <li>• Communication line to IMS</li> <li>• DD names for polled devices</li> <li>• LTERM definition</li> <li>• System/3 and System/7 line groups</li> <li>• Physical characteristics of terminal</li> </ul>
Switched Communication Device	CONFIG Macro IDLIST Macro LINE Macro LINEGRP Macro NAME Macro POOL Macro SUBPOOL Macro TERMINAL Macro	<ul style="list-style-type: none"> <li>• Switched 3275 configuration</li> <li>• Security for switched 3275</li> <li>• Communication line to IMS</li> <li>• DD names for polled devices</li> <li>• LTERM definition</li> <li>• Set of LTERMs for switched line group</li> <li>• Set of LTERMs for a switched line terminal</li> <li>• Physical characteristics of terminal</li> </ul>
Multiple System Coupling	MSLINK Macro MSNAME Macro MSPLINK Macro NAME Macro	<ul style="list-style-type: none"> <li>• Logical Link names</li> <li>• Partner systems</li> <li>• Physical link names</li> <li>• LTERM definition</li> </ul>
VTAM	COMM Macro NAME Macro SUBPOOL Macro TERMINAL Macro TYPE Macro VTAMPOOL Macro	<ul style="list-style-type: none"> <li>• General communication options</li> <li>• Set of LTERMs for a static ISC session</li> <li>• Physical characteristics of a static terminal</li> <li>• VTAM terminal type</li> <li>• Multiple systems with ISC sessions</li> </ul>

Table 18. System Definition Macros and Their Uses

Type of Macro Statement	Macro Name	Resources Each Macro Defines
<b>System Configuration Macros</b>	"BUFPOOLS Macro" on page 67	Buffer pool sizes
	FPCTRL Macro	Fast Path options
	IMSCTF Macro	System control integrity factors
	IMSCTRL Macro	Library naming, JCL options, and ETO feature
	IMSGEN Macro	z/OS system and control options
	MSGQUEUE Macro	Message queue device and message sizes
	SECURITY Macro	Security options
<b>Database and Application Macros</b>	APPLCTN Macro	Program characteristics and PSB names
	DATABASE Macro	DBD names
	RTCODE Macro	Fast Path message routing
	TRANSACT Macro	Transaction processing options

Table 18. System Definition Macros and Their Uses (continued)

Type of Macro Statement	Macro Name	Resources Each Macro Defines
Data Communication Macros	BTAM, BSAM, GAM, ARAM	COMM Macro General communication options
		CTLUNIT Macro 2848, 2972, 3271 control
		LINE Macro Communication line to IMS
		LINEGRP Macro DD names for polled devices
		NAME Macro LTERM definition
		STATION Macro System/3 and System/7 line groups
Switched Communication Device	TERMINAL Macro	Physical characteristics of terminal
	CONFIG Macro	Switched 3275 configuration
	IDLIST Macro	Security for switched 3275
	LINE Macro	Communication line to IMS
	LINEGRP Macro	DD names for polled devices
	NAME Macro	LTERM definition
	POOL Macro	Set of LTERMs for switched line group
	SUBPOOL Macro	Set of LTERMs for a switched line terminal
Multiple System Coupling	TERMINAL Macro	Physical characteristics of terminal
	MSLINK Macro	Logical Link names
	MSNAME Macro	Partner systems
	MSPLINK Macro	Physical link names
VTAM	NAME Macro	LTERM definition
	COMM Macro	General communication options
	SUBPOOL Macro	Set of LTERMs for a static ISC session
	TERMINAL Macro	Physical characteristics of a static terminal
	TYPE Macro	VTAM terminal type
	VTAMPOOL Macro	Multiple systems with ISC sessions

## Guidelines for IMS Macros

This chapter presents the IMS system definition macro statements in alphabetic order. Each macro statement description contains a figure showing the macro's syntax and an explanation of the macro's keywords and parameters.

"Resource Naming Rules" on page 60 describes rules and restrictions that apply to IMS macros; "Coding Conventions" on page 62 describes coding conventions for macros and examples.

## Resource Naming Rules

These rules and restrictions apply to all IMS macros.

- Names cannot include a blank, comma, period, hyphen, or equal sign.
- All PSB names must begin with an alphabetic character (alphabetic = A through Z, #, \$, and @) followed by zero to seven alphanumeric characters.
- Logical terminal names and transaction codes must be one to eight alphanumeric characters in length. (Alphanumeric = A through Z, #, \$, and @, 0 through 9.) Each name within the set comprising LTERM names and transaction codes must be unique.
- User IDs that are passed to RACF must be one-to eight-alphanumeric characters, that is, letters A through Z, digits 0 through 9, and the national characters #, \$, and @.

For ETO terminals, if the user ID is used to build the LTERM structure, the user ID must be one-to eight-alphanumeric characters.

- Node names must be unique among themselves, but can duplicate LTERM names, transaction codes, or subpool names.
- Subpool names must be unique among themselves, but can duplicate LTERM names, transaction codes, or node names.
- IMS null words **must not** be used as resource names (FOR, TO, ON, AFTER, and SECURITY). Also, resource names **must not** begin with DFS (except for DFSIVPxx and DFSSAMxx) or DBCDM; nor should WTOR, MSDB, SDB, DBRC, BASICEDT, or ISCEDT and its aliases be used as a resource name.
- Command keywords must not be used as resource names.

**Related Reading:** See *IMS Version 9: Command Reference* for more information on command keyword restrictions.

- Each IMS macro statement can appear in an IMS system definition a limited number of times. Table 19 shows the maximum number of times each macro statement can occur.
- Symbolic names can be used to define 3270 and SLU Type 2 devices. See the description of the TYPE= keyword of the TERMINAL macro statement for the format of these names. A symbolic name must be used to define the format of any 3270 display device with a screen size other than 480 or 1920 characters.
- The maximum number of IMS resources that can be defined is a design limit (that is, a theoretical limit). You should not assume that your installation can successfully define the maximum number of IMS resources (listed in Table 19). The actual limit on this number is influenced by many limiting factors in your hardware and software configuration.

Table 19. Maximum Occurrences of Each IMS System Definition Macro Statement

Macro Statement	Maximum Occurrences/ Specification the assembler
APPLCTN	999,999
BUFPOOLS	1
COMM	1
CONFIG	255
CTLUNIT	1000
DATABASE <sup>1</sup>	32,700
FPCTRL	1
IDLIST	1000
IMSCTF	1

Table 19. Maximum Occurrences of Each IMS System Definition Macro Statement (continued)

Macro Statement	Maximum Occurrences/ Specification the assembler
IMCTRL	1
IMSGEN	1
LINE	1000 - (Number of POOL macros specified)
LINEGRP	676 - (Number of MSPLINK macros specified)
MSGQUEUE	1
MSLINK	675
MSNAME	676
MSPLINK	676 - (Number of LINEGRP macros specified)
NAME	999,999 - (Number of switched line macros specified)
POOL	1000 - (Number of LINE macros specified)
RTCODE	999,999
SECURITY	1
STATION	999,999 - (Number of TERMINAL, SUBPOOL, and CONFIG macros specified)
SUBPOOL	999,999 - (Number of TERMINAL, STATION, CONFIG macros specified)
TERMINAL <sup>2</sup>	145,000 - (Number of SUBPOOL, CONFIG, and STATION macros specified)
TRANSACT	999,999
Transaction Edit Routines <sup>3</sup>	255
TYPE	999,999
User routines <sup>4</sup>	200
VTAMPOOL <sup>5</sup>	1

**Notes:**

1. In IMS execution, database allocation is limited to the number of ddnames that z/OS allows.
2. Because of the size of the IMS control block, the maximum number of static terminal resources that can be defined is approximately 145,000. If your system includes a large number of terminals, IBM recommends that you use the ETO feature. The ETO feature enables you to manage terminals more effectively during IMS online activity.
3. Transaction Edit routines are specified in the EDIT operand of the TRANSACT macro. Although the limit of TRANSACT statements is 999,999, only 255 transaction edit routines are accepted.
4. *User Routines* refer to the user-supplied physical terminal output and input edit routines (specified in the EDIT operand of the LINEGRP and TYPE macros).
5. VTAMPOOL is used once for each set of LU 6.1 subpools defined. It can be used multiple times within a system definition to define multiple sets of LU 6.1 subpools.

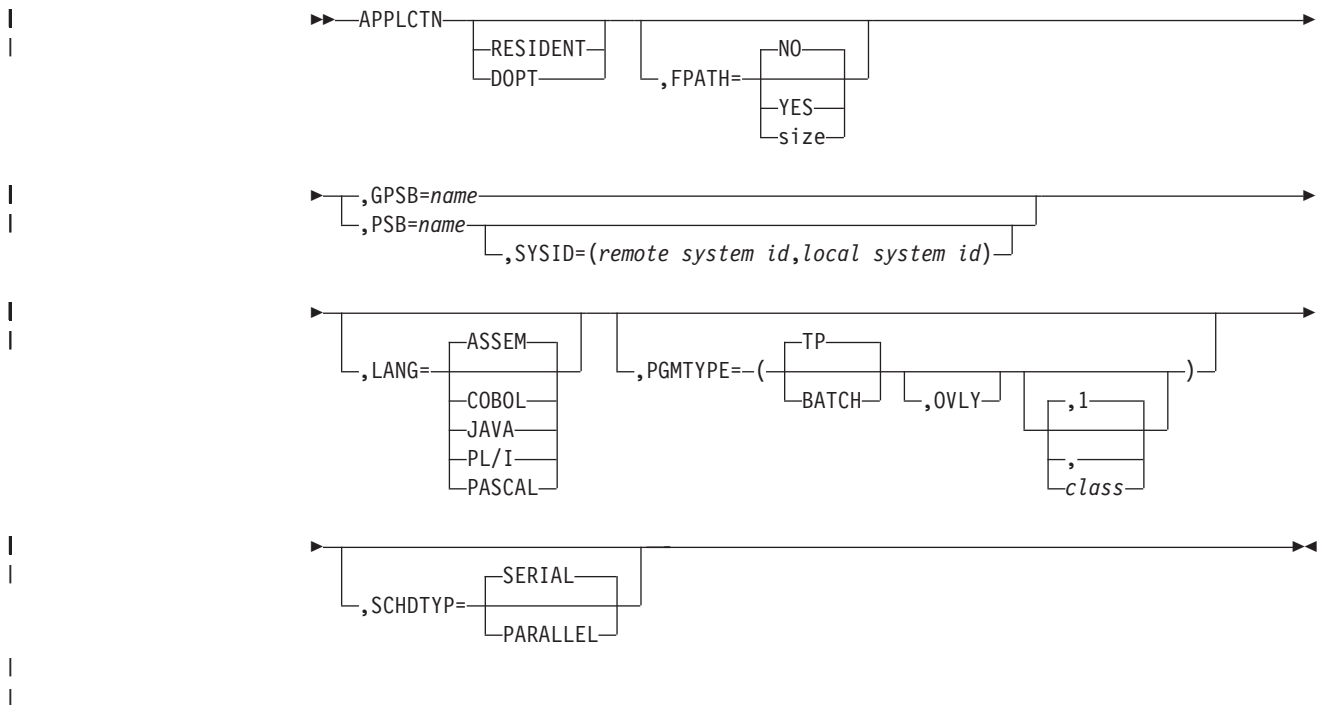
## Coding Conventions

Both this book and its co-requisite, *IMS Version 9: Installation Volume 1: Installation Verification*, use the following conventions in coding macros and examples:

- Uppercase letters, stand-alone numbers, and punctuation marks must be coded exactly as shown. The only exceptions are subscripts, which are not coded.
- Lower-case letters, words, and associated numbers represent variables for which specific values can be substituted.
- A space or the lower-case “b” indicates one blank position.
- If an alternative item is underlined, it is the default. IMS assumes that if none of the items is coded, the underlined item is automatically the choice.
- You must specify positional parameters in the order shown.
- You can specify keyword parameters in any order.

## APPLCTN Macro

The APPLCTN macro allows you to define the program resource requirements for application programs that run under the control of the IMS DB/DC environment, as well as for applications that access databases through DBCTL. An APPLCTN macro combined with one or more TRANSACT macros defines the scheduling and resource requirements for an application program. Using the APPLCTN macro, you only describe programs that operate in message processing regions, Fast Path message-driven program regions, batch message processing regions, or CCTL threads. You do use the APPLCTN macro to describe application programs that operate in batch processing regions. When defining an IMS data communication system, at least one APPLCTN macro is required.



## Positional Parameters

The positional parameter `RESIDENT` specifies that the PSB associated with this application program is to be made resident during system initialization. `RESIDENT`



and DOPT (dynamic PSB option) are mutually exclusive. DOPT and SCHDTYP=PARALLEL are also mutually exclusive.

DOPT specifies that the PSB associated with this application program is to be located dynamically. If DOPT is specified, the following actions are taken during the execution of the IMS control region:

- Initialization does not perform a BLDL on the PSB associated with this application program. Thus the PSB does not need to be in any data set defined by the ACBLIB DD statement until it is actually required to process a transaction. This is not true, however, of the DBDs that the PSB might reference. All DBDs that are to be used online must be available during initialization in a data set defined by the ACBLIB DD statement. PSBs referencing DBDs that are missing during initialization are not scheduled.
- Each time the program associated with this PSB is scheduled, a BLDL is performed, and the latest copy of the PSB is located. A BLDL is not performed for associated DBDs, so the DBD cannot be modified until the system is initialized again or until online change is used.
- When the program terminates, the PSB is deleted from the PSB pool as part of the termination process.

Neither RESIDENT nor DOPT is the default parameter. Rather, if neither RESIDENT nor DOPT is selected on the APPLCTN macro statement, IMS system initialization causes a BLDL to be performed on the PSB associated with the application being defined. The PSB is not made resident (that is, loaded from the ACBLIB) until the application is scheduled.

To provide greater flexibility in the user's control of the library containing the dynamic PSBs, the following restrictions are imposed:

- The PSB must reside in a library other than the primary ACBLIB and must be concatenated to it.
- The concatenated library containing the PSB must be in ACBLIB format.
- The IMS.PSBLIB data set cannot be used.

If the BLDL performed at the scheduling of the PSB determines that the PSB resides in the first concatenation of the IMSACB DD statement set, the PSB is stopped and an error message is sent to the master terminal. The PSB is not scheduled. If the dynamic PSB is added to the concatenation of the IMS.ACBLIB, causing the data set to be expanded to a secondary extent, that PSB is not available to the online system until the ACBLIB is closed and reopened by IMS.

For MSC remote applications, the APPLCTN macro provides documentation and a reference for its transactions. It does not generate a control block. The PSB need not be generated in systems where it is used as a remote reference. The PSB name is defined here for documentation purposes.

To dynamically reassign a transaction from remote to local processing, a control block must be present in the local system.

**Related Reading:** See the description of the /MSASSIGN command in *IMS Version 9: Command Reference* for additional information on dynamic reassignment.

Thus, if dynamic reassignment is desired, a transaction must be defined as local. That is, it must be under a local application having the same PSB name as the

remote application (using the APPLCTN macro without a SYSID parameter). If dynamic reassignment is not desired, local definition is unnecessary.

Online changes to both the RESIDENT and DOPT options can be made via a MODBLKS system definition. However, PSBs associated with new application programs defined as RESIDENT (that is, changed from NONRESIDENT or DOPT to RESIDENT) during a MODBLKS system definition are not made resident until the next restart of IMS. Until that time, they are to be treated as nonresident. Online changes made to already-resident PSBs (via an ACBGEN) cause the PSB to be treated as nonresident until the next IMS restart.

MPPs scheduled against dynamic PSBs are not allowed to go through the quick reschedule process or become pseudo WFIs.

**Related Reading:** For additional information on PSBs defined with DOPT see *IMS Version 9: Utilities Reference: System* under "Application Control Blocks (ACB) Maintenance Utility."

## Keyword Parameters

To find which parameters apply to your IMS configuration, refer to Table 15 on page 33.

### FPATH=

Specifies whether (YES) or not (NO) this is a Fast Path-exclusive application program. FPATH=size, which determines the EMH buffer size required to run the transaction, overrides the EMHL execution parameter and implies FPATH=YES. The minimum specification for FPATH=size is 12; the maximum is 30720. FPATH=YES implicitly defines a wait-for-input (WFI) application program. The PGMTYPE= parameters that define the overlay structure and class are invalid if FPATH=YES is specified. The SYSID= parameter is also invalid if FPATH is specified.

If FPATH=YES is specified during a MODBLKS system definition, Fast Path must have been previously defined for the online system to which this change is made.

Fast Path-potential transactions must be able to run under two applications. One of these applications must be defined with FPATH=YES; the other with FPATH=NO. The application with FPATH=YES must also be defined with either the same transaction or with a routing code that can be assigned by the user Input Edit/Routing exit routine.

FPATH=NO must be specified if specifying LANG=JAVA.

**Related Reading:** See the TRANSACT macro in this book, and also the *IMS Version 9: Customization Guide* for information about the User Input Edit/Routing exit routine.

When Fast Path is included during system definition, the FPBUF parameter on the TERMINAL macro statement is ignored, except to determine the default EMH buffer size. Fast Path buffers are provided by the EMHB pool, which expands and contracts dynamically depending on the number of ETO terminals concurrently entering Fast Path transactions.

### GPSB=

Causes the scheduling process of all environments to generate a PSB containing an I/O PCB and an alternate modifiable PCB. With the GPSB= keyword, you do not need to perform the PSBGEN and ACBGEN, thus eliminating I/O to the ACBLIB.

GPSB= generates an I/O PCB named IOPCBbbb; The modifiable, alternate PCB is named TPPCB1bb. With an alternate modifiable PCB, an application can use the CHNG call to change the output destination and send output to a destination other than the input destination.

You can make an online change to add the GPSB option to an existing application, or add a new application with the GPSB options by using a MODBLKS system definition. However, the GPSB option does not take effect unless the ACBLIB is also changed with online change.

If LANG=JAVA is specified and the Java application is a message processing program, the name specified on GPSB= is the name of the Java application class.

### **LANG=**

Defines the language interface of the application program. You can use the following values with the LANG= keyword:

ASSEM is the default value.

If GPSB= is specified, LANG= is used to determine the language; it is used only with GPSB=.

If LANG=JAVA is specified, then FPATH=NO must also be specified.

### **PGMTYPE=**

Specifies application program characteristics. The first parameter describes the type of application program being defined.

#### **TP**

Specifies that IMS schedules the program when messages processed by the program exist in the system. This is the default.

#### **BATCH**

Can use DL/I in the IMS control program system region and can refer to the message queues. If BATCH is coded, all TRANSACT macro statements that follow are assigned normal and limit priority values of zero.

#### **OVLY**

No longer used by IMS but is retained for compatibility. An execution time parameter, OVLA on procedure DFSMPR, is available when starting an MPP to indicate whether the overlay supervisor should be preloaded by IMS.

If a program is changed from batch to online using the online change facility, you must enter the /ASSIGN command to assign nonzero current, limit, and normal priorities (CPRI, NPRI, and LPRI keywords) to the transactions using that program. This is because the online change facility does not alter attributes that are changeable through the /ASSIGN command. Regardless of whether you specify new values for the transaction in the MODBLKS system definition, they are ignored during /MODIFY processing.

If FPATH=YES is specified:

- TP specifies a message-driven Fast Path application program
- BATCH cannot be specified. If BATCH is specified, an error message is issued. Fast Path nonmessage-driven application programs are not supported, and should be changed to run as BMPs.
- The PGMTYPE= parameters that define the overlay structure and class are not valid.
- The SYSID= parameter is not valid.

The third parameter of the PGMTYPE= keyword specifies the class to which the transaction codes specified in the following TRANSACT macro statements are to be assigned. This parameter must be a decimal number from 1 to 999. This value must not exceed the value given (by specification or default) on the MAXCLAS= keyword of the IMSCTRL macro. The default is 1. If the transaction code class is to be specified in the individual TRANSACT macro statements, this parameter need not be coded. If the transaction code class is specified in both the APPLCTN and TRANSACT macro statements, the APPLCTN macro specification is ignored, and the TRANSACT macro specification is used.

If the PGMTYPE= (,class) parameter is to be changed online, the class value that is specified cannot exceed the definition (by specification or default) on the MAXCLAS= keyword of the IMSCTRL macro in the online system to which this change is to be made.

The numeric class subparameter must not be specified if FPATH=YES is specified.

**PSB=**

Specifies the name of the PSB associated with this application program definition. Each local PSB name must be unique. A remote and a local application can each be defined as having the same PSB name. This is required in order to dynamically reassign a transaction from remote to local processing. The first character of the PSB name must be a letter. If PGMTYPE=TP, the PSB name must also be the program name.

**Related Reading:** For the description of the /MSASSIGN command, see *IMS Version 9: Command Reference*.

**SYSID=**

Specifies in the multiple-IMS system configuration, the system identification (SYSID) of the remote system (that system on which the application executes) and the SYSID of the local system (the originating system to which the responses are returned). The values specified must be numbers in the range from 1 to 2036.

The remote SYSID specified must also be defined in an MSNAME macro statement, but the local SYSID can be defined in any or all of the MSNAME, TRANSACT, and APPLCTN macro statements. If SYSID is specified, all other keywords except PSB are ignored.

If the SYSID parameter is specified in the APPLCTN macro statement, you need not specify the SYSID in the TRANSACT macro statement. If the SYSID is specified in both the APPLCTN and TRANSACT macro statements, the APPLCTN specification is ignored.

The SYSID parameter is independent of the link type (BSC, CTC, MTM, VTAM) specified on the TYPE= keyword of the MSPLINK macro statement.

Because the values associated with the SYSID= keyword can be changed using an /MSASSIGN command, you should not add SYSID= to an existing APPLCTN macro for a MODBLKS system definition. Doing so causes the local application to be deleted after an online change sequence of commands.

A PSB cannot be initially defined or redefined from remote to local by using the SYSID= parameter on the APPLCTN macro during an online change.

The SYSID keyword parameter is invalid if FPATH=YES is specified.

**SCHDTYP=**

Specifies whether (PARALLEL) or not (SERIAL) this application program can be

scheduled into more than one message region or batch message region simultaneously. The default value is SERIAL.

When the SCHDTYP= parameter is changed by a MODBLKS system definition from serial to parallel or vice versa, and the PSB, defined as resident, is not changed, the PSB is considered non-resident until the next IMS restart.

|  
|

SCHDTYP=PARALLEL and the positional parameter DOPT are mutually exclusive.

## BUFPOOLS Macro

The BUFPOOLS macro statement is used to specify default storage buffer pool sizes for the DB/DC and DBCTL environments. The sizes specified are used unless otherwise expressly stated for that buffer or pool at control program execution time for an online system.

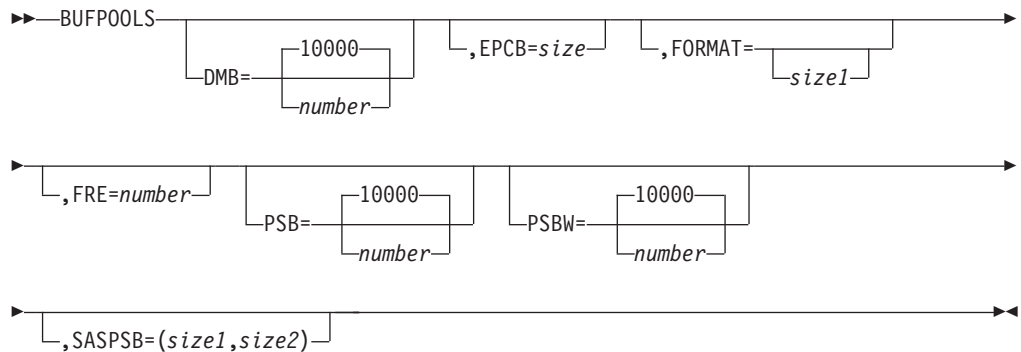
The following keywords are no longer supported:

- The DBASE keyword.  
You now specify the common database buffer pool size using DFSVSAMP data set parameters.
- The COMM keyword.
- The GENERAL keyword.

Values entered for these keywords are ignored. The DBASE keyword, which specified the size of the common database buffer pool, is no longer supported. The common database buffer pool size is now specified using DFSVSAMP data set parameters. See "Specifying IMS Buffer Pools" on page 430 for more information on common database buffer pool definition.

The COMM keyword, which specified additional space to be allocated to the CIOP pool, is no longer supported. The GENERAL keyword, which specified the size of the SPAP pool, is also no longer supported. Values entered for either of these keywords are ignored.

The Work Area Pool (WKAP) cannot be specified at system definition time. System definition sets a value of 5000. You can override this at execution time by using the WKAP= operand in the IMS, DBC, or DCC procedures, or in the DFSPBIMS, DFSPBDBC, or DFSPBDCC members.



## Keyword Parameters

To find which parameters apply to your IMS configuration, refer to Table 15 on page 33.

**DMB=**

Specifies the size of the DMB control block pool. The default is 10000 bytes. The maximum allowable specification is 9999000 bytes. The minimum allowable specification is 8 bytes.

**EPCB=**

Specifies the size of the EPCB pool. If Fast Path is generated, the default size is 8 KB. Otherwise the default EPCB pool size is 0. The maximum allowable specification is 9999000 bytes. The storage for this pool resides in ECSA.

Table 20 shows how to calculate EPCB storage for PSBs. The calculation is for MPP, IFP, and BMP region types that use Fast Path resources (DEDBs, MSDBs, or EMH).

Table 20. Calculating EPCB storage for PSB

EPCB Type	Size (in bytes)	Total VS Requirements for This EPCB Type	Total
IOPCB	28 b	28	_____K
Alt-Resp	28 b	28 x (# Alt-Resp PCBs)	_____K
MSDB	76 b	76 x (# of MSDB PCBs)	_____K
DEDB	* b	Sum of all DEDB PCBs	_____K
Total EPCB storage for this PSB			_____K

**Formula for Calculating Value of EPCB Pool:**

132 + (124 x # of SENSEG statements in the DEDB PCB).

After working this calculation for each PSB, determine the largest PSB that might execute in a region, and add up all the results for all regions. This is the value for your EPCB pool. This pool specification can be overridden via the IMS procedure using the EPCB parameter.

**FORMAT=**

Specifies sizes of the message format buffer pool.

*size1* specifies the size of the message format buffer pool. The minimum specification is 2000 bytes; the maximum specification is 99999000 bytes. This parameter is ignored when no unit types requiring MFS are included in the defined system. If unit types that require MFS are included in the defined system and this parameter is not specified, the default value is calculated. *size2* is no longer supported.

The default value is calculated as follows:

$$1000 + X + 2124 + FRE \times 44$$

where:

**X** Is the greater of: ((3270B/2) + (3270V/8)) x 2200 or 14336

**3270B** Is the number of 3270 BTAM lines

**3270V** Is the number of 3270 VTAM terminals

**FRE** is the number of specified or defaulted FREs

**FRE=**

Specifies the number of fetch request elements (FREs) for loading MFS control blocks into the message format buffer pool. The number specified can range

from 10 to 99999. If the size of the buffer pool is increased, the number of FREs should also be increased. Otherwise, the additional space cannot be used.

The default value is calculated as follows:

$(10 + X)$  or 30, whichever is greater

where:

**X**  $((3270B/2) + (3270V/8)) \times 4$

**3270B** Is the number of 3270 BTAM lines

**3270V** Is the number of 3270 VTAM terminals

#### **PSB=**

Specifies the size of the PSB control block pool if the DL/I address space option is not used. The SASPSB parameter specifies the size of the PSB control block pool when the DL/I address space option is used. The default is 10000 bytes, with a maximum of 9999000 bytes. The minimum allowable specification is 8 bytes.

#### **PSBW=**

Specifies the size of the PSB work area pool. The default is 10000 bytes. The maximum allowable specification is 9999000 bytes. The minimum allowable specification is 8 bytes.

#### **SASPSB=**

Is used only if the DL/I subordinate address space option is selected. If you are not using this option, the size of the single PSB control block pool is specified with the PSB parameter.

With the DL/I address space option, two PSB control block pools exist. *Size1* is the size of the pool in the OS/390 CSA. *Size2* is the size of the pool in DL/I local storage. The maximum allowable for either is 9999000 bytes.

Both *size1* and *size2* must be specified. Normally, the value of *size2* should be larger than that of *size1*, and neither value can be 0.

The ACBGEN utility output provides information on the relative PSB pool sizes. You should examine the output of ACBGEN before coding SASPSB.

The defaults are:

*size1* 20% of the PSB specification on BUFPOOLS

*size2* 80% of the PSB specification on BUFPOOLS

For example, SASPSB=(20000) is invalid, because *size2* is missing. The defaults apply only if the SASPSB parameter is not specified.

**Example:** The following example shows how to code BUFPOOLS. If the DL/I address space option is not used, the single PSB pool size is 80000 bytes. If the option is selected, the PSB pool size in the CSA is 16000 bytes, and the DL/I local storage pool size is 64000 bytes. This example also specifies a 40000-byte DMB pool.

```
BUFPOOLS PSB=80000,SASPSB=(16000,64000),
DMB=40000
```

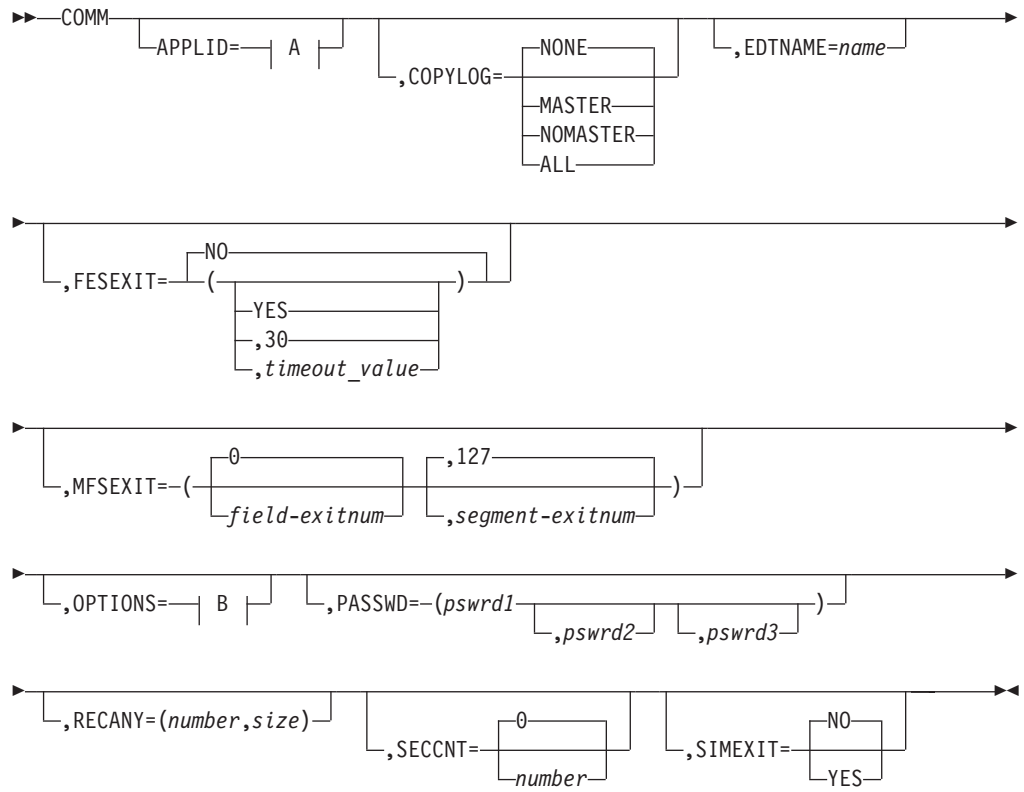
---

## COMM Macro

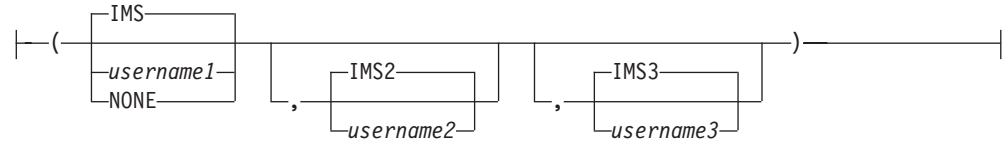
The COMM macro is used to specify general communication requirements that are not associated with any particular terminal type. COMM is always required for terminal types supported by VTAM. It is optional for BTAM, BSAM, GAM, and ARAM terminal types. It can also be required to specify additional system options, such as support for MFS on the master terminal.

The COMM macro should be placed prior to the data communication specifications in the stage 1 input sequence. Refer to Appendix B, "MFS System Definition and Programming Considerations," on page 523 for specific examples.

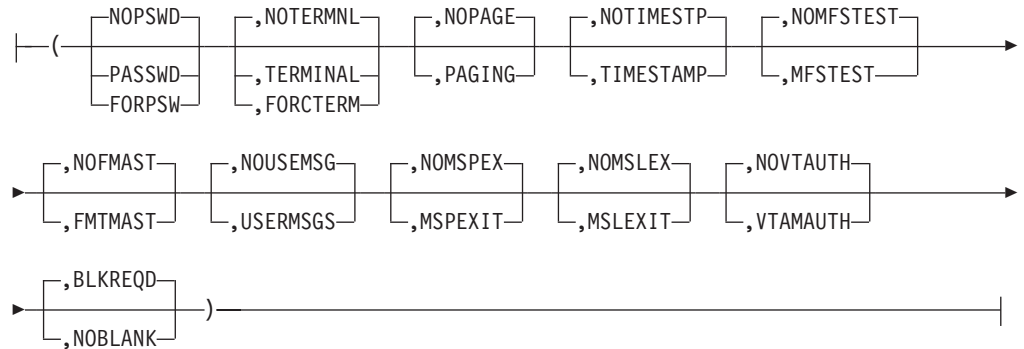




**A:**



**B:**



### Keyword Parameters

To find which parameters apply to your IMS configuration, refer to Table 15 on page 33.

**AOEXIT=**

No longer supported.

**APPLID=**

Specifies the application identification in the VTAM ACBs if VTAM is used. The default value is IMS.

For XRF, specify two VTAM APPLIDs: *username1* and *username2*. These two parameters correspond to the two XRF systems. The defaults are IMS and IMS2, respectively. When MNPS is used for XRF, *username1* and *username2* refer to the APPLID ACBs in the two XRF systems.

For RSR without XRF, you should specify at least two VTAM APPLIDs: *username1* (or *username2*) and *username3*, where *username3* is the application identification for the RSR tracker. In all cases, *username3* can only be used by the RSR tracker.

For RSR with XRF, you use all three VTAM APPLIDs: *username1* is for your active IMS (for both XRF and RSR), *username2* is your XRF alternate (at the RSR active site), and *username3* is your RSR tracker.

All three username values can be overridden by specifying APPLID1=, and either APPLID2= or APPLID3=, or both, in the IMS or DCC procedures. These keywords can also be specified in the DFSPBIMS or DFSPBDCC members. *Username3* can match either *username1* or *username2*, but no two APPLIDs can be active in the network at the same time.

The defaults are IMS, IMS2, and IMS3, respectively.

If NONE is specified:

- The MSVERIFY program (DFSUMSV0) cannot validate that the APPLID names of the IMS system and its partner match if MSC/VTAM links are defined.
- Either the job step name of the job to bring up the IMS control region is used, or the name of the started task (if a started task is used for the IMS control region) is used.

If you specify APPLID=NONE and you issue a /DIS ACTIVE DC command, the output of the command is APPLID=NONE.

APPLID=NONE cannot be specified for XRF or for RSR.

**COPYLOG=**

Specifies the different categories of command and command response messages for which a copy is to be printed on the secondary master terminal.

If you want hardcopy logging, you must define a secondary master terminal.

The default is NONE.

**NONE** Specifies that the commands listed in Table 21 on page 73 and command responses from terminals are not to be printed on the secondary master terminal. The queuing for the secondary master terminal of messages not in the categories described in Table 21 on page 73 is not affected by the COPYLOG statement.

**MASTER**

Specifies that copies of commands listed in Table 21 on page 73 and command responses from the master terminal are to be printed on the secondary master terminal.

**NOMASTER**

Specifies that copies of the commands listed in Table 21 and command responses from terminals other than the master terminal are to be printed on the secondary master terminal.

**ALL**

Specifies that the commands listed in Table 21 and command responses from all terminals are to be printed on the secondary master terminal.

COPYLOG has no effect on system messages, which continue to be logged to the secondary master terminal.

The commands that can be copied to the secondary master terminal are:

*Table 21. Commands That Can Be Copied to the Secondary Master Terminal*

/ACTIVATE	/MSASSIGN
/ASSIGN	/OPNDST
/CHECKPOINT	/PSTOP
/CLSDST	/PURGE
/COMPT	/QUIESCE
/DBDUMP	/RCLSDST
/DBRECOVERY	/RCOMPT
/DELETE	/RSTART
/DEQUEUE	/START
/DISPLAY	/STOP
/IDLE	/TRACE
/MONITOR	/UNLOCK SYSTEM

Other commands can be logged but are not under the control of COPYLOG as /MSVERIFY is, for example.

The COPYLOG specification can be dynamically changed at execution time by the /SMCOPY operator command.

**EDTNAME=**

Names the ISC edit process for LU 6.1 nodes. The name specified becomes a synonym for ISCEDT, the default name that can be input and is output on the DPN parameter of the SCHEDULER function management header. When this name is input to IMS on the DPN parameter of the ATTACH or SCHEDULER function management header, it causes IMS to use the ISC edit process. The name used on the EDTNAME parameter cannot be the same as any MFS message input descriptor (MID) name used as an input ATTACH DPN parameter to invoke an MFS process.

**FESEXIT=**

Specifies whether (YES) or not (NO) the Front End Switch exit routine is to be included in the IMS nucleus.

The default is NO.

The *timeout\_value* specifies a timeout value in seconds. A *timeout\_value* specifies the time interval in which a reply for an FES message must be received. If no reply is received by the front-end system in time, timeout processing begins. The minimum value is 1, and the maximum value is 300.

If FESEXIT=YES is specified without a timeout value, the default for the *timeout\_value* parameter is 30.

The FES exit routine must be placed in the data set defined by the USERLIB= keyword (of the system definition MSGEN macro) prior to the execution of stage 2. The name of the exit routine must be DFSFEBJ0.

FESEXIT=(YES) and OPTIONS=NOBLANK are mutually exclusive. If you specify both, stage one assembly will not complete successfully.

**Related Reading:** More information on the Front End Switch exit routine can be found in *IMS Version 9: Customization Guide*.

#### MFSEXIT=

The variable *fieldexitnum* specifies the highest-numbered MFS Field Edit exit routine to be included in the generated system. *fieldexitnum* must be a decimal number greater than or equal to 0 and less than the default or specified value for the *segmentexitnum* parameter. The default is 0.

The *segmentexitnum* value specifies the lowest-numbered MFS Segment Edit exit routine to be included in the generated system. The *segmentexitnum* must be a decimal number greater than the default or specified value for the *fieldexitnum* parameter and less than or equal to 127. The default is 127.

**Recommendation:** Make lower-numbered exit routines be field exit routines, even though the lower-numbered exit routines can be segment exit routines, or can be a mixture of field and segment exit routines. An additional recommendation is that the higher-numbered exit routines be segment exit routines, although the higher-numbered exit routines can be field exit routines, or can be a mixture of field and segment exit routines.

An MFSEXIT=(14,120) specification indicates that user-supplied MFS Field Edit exit routines DFSME001 through DFSME014 and user-supplied MFS Segment Edit exit routines are IMS-supplied routines and are always included in the generated system when MFS requirements exist.

User-supplied MFS exit routines are included in the system definition-generated online control blocks module DFSIBLKx. Therefore, MFS exit routine specifications can be changed via an IMS CTLBLKS, NUCLEUS, ON-LINE, or ALL system definition. The MFSEXIT keyword specification is ignored when MFS exit requirements do not exist in the generated system.

#### OPTIONS=

The parameters included in this description are not position dependent within the operand sublist. If any terminal or password security options are specified here and also in the IMSGEN or SECURITY macro statement, a warning message will be issued.

**Related Reading:** See “Relationships between the SECURITY, COMM, and IMSGEN Macros” on page 136 for more information about security options.

#### **NOPSWD** | **PASSWD** | **FORPSW**

Specifies whether (PASSWD or FORPSW) or not (NOPSWD) password security is to be generated as the default for the /NRESTART command when issued to start IMS. This parameter value is ignored if the SECURITY macro is specified, and it is included for compatibility purposes only. Default value is NOPSWD. See “Relationships between the SECURITY, COMM, and IMSGEN Macros” on page 136 for more information.

FORPSW specifies that the IMS master terminal operator cannot negate password security when issuing the /NRESTART command. If PASSWD is specified, the MTO can override password security inclusion by specifically requesting no security on the /NRESTART command.

#### **NOTERMNL** | **TERMINAL** | **FORCTERM**

Specifies whether (TERMINAL or FORCTERM) or not (NOTERMNL) terminal security is to be generated as the default for the /NRESTART command when issued to start IMS. This parameter will be ignored if the

SECURITY macro is specified, and is included for compatibility purposes only. The default is NOTERMNL. See “Relationships between the SECURITY, COMM, and IMSGEN Macros” on page 136 for more information.

FORCTERM specifies that the IMS MTO cannot negate terminal security when issuing the /NRESTART command. If TERMINAL is specified, the master terminal operator can override terminal security by specifically requesting no security on the /NRESTART command.

#### **NOPAGE | PAGING**

Defines whether (PAGING) or not (NOPAGE) the terminal paging feature is included in the defined system. The default is NOPAGE. If 3270 or SLU 2 devices are included, or if SLU P, LU 6.1, or 3600 devices defined with MFS are included, the paging capability is automatically included.

#### **NOTIMESTP | TimestAMP**

Specifies whether (TimestAMP) or not (NOTIMESTP) the system message time stamp feature is desired in the defined system. The default is NOTIMESTP.

When TimestAMP is specified, the time at which a message was generated is inserted between the message number and the message text for each message in the table.

**Related Reading:** Refer to the DFS messages introduction in *IMS Version 9: Messages and Codes, Volume 2* for more information.

#### **NOMFSTEST | MFSTEST**

Specifies whether (MFSTEST) or not (NOMFSTEST) the Message Format Service test facility, MFSTEST, is to be included in the generated system. Specifying MFSTEST is invalid for systems not containing MFS terminals. The default is NOMFSTEST.

Specifying a numeric value on the COMM macro statement is no longer recommended; for compatibility reasons it is still allowed, but any numeric value is ignored. When a numeric value is specified the NOMFSTEST default is overridden and MFSTEST is set; if MFSTEST=NO is specified on the IMSGEN macro, it is also overridden.

Using MFSTEST can degrade IMS performance due to MFSTEST use of the CIOP pool.

#### **NOFMAST | FMTMAST**

Specifies whether (FMTMAST) or not (NOFMAST) the IMS-provided support for MFS on the master terminal is to be used. This support is available for the following devices used as the master terminal:

- 3277 model 2
- 3270 displays that have a 24 x 80 screen and use symbolic names
- 3277 model 2 when defined as SLUTYPE2

#### **NOUSEMSG | USERMSGs**

Specifies whether (USERMSGs) or not (NOUSEMSG) the user-supplied user message tables module DFSCMTU0 is to be included within the generated system.

#### **NOMSPEX | MSPEXIT**

Specifies whether (MSPEXIT) or not (NOMSPEX) the user-written MSC Program Routing exit routine (DFSCMPRO) is to be included in the generated system.

**NOMSLEX | MSLEXIT**

Specifies whether (MSLEXIT) or not (NOMSLEX) the user-written MSC Link Receive Routing exit routine (DFSCMLR0) is to be included in the generated system.

**NOVTAUTH | VTAMAUTH**

Specifies whether (VTAMAUTH) or not (NOVTAUTH) IMS is to use the VTAM authorized path for communicating.

**Related Reading:** Refer to *ACF/VTAM Macro Language Guide* for information about the authorized path facility. These parameters can be overridden by the JCL prepared for system execution.

**BLKREQD | NOBLANK**

Specifies whether the one- to eight-byte transaction code requires a trailing blank. NOBLANK lets you enter a one- to eight-byte transaction code without having to specify a trailing blank.

OPTIONS=NOBLANK and FESEXIT=(YES) are mutually exclusive. If you specify both, stage one assembly does not complete successfully.

**PASSWD=**

Specifies, when neither XRF nor RSR is used, that one password is to be specified in the VTAM ACB. This password is checked by VTAM. If no password is specified, either in the VTAM ACB or in the PASSWD1= parameter, and VTAM requires one (during VTAM system definition), the IMS VTAM ACB is not initialized.

For XRF, you should specify two VTAM passwords that correspond to the two XRF systems. If only one password is specified, it is used as the password for both XRF systems.

For RSR without XRF, you should specify at least two VTAM passwords:

- *password1* (or *password2*)
- *password3*

In all cases, *password3* can only be used by the RSR tracker.

For RSR with XRF, you will use all three VTAM passwords:

- *password1* for your active IMS (for both XRF and RSR)
- *password2* for your XRF alternate (at the RSR active site)
- *password3* for your RSR tracker

If you specify only one password, it is used for all systems. If you specify only two passwords, *password1* is used for the RSR tracker.

**RECANY=**

This parameter is required only when defining VTAM terminals. It defines the number and size of the RECEIVE ANY buffers.

This keyword and its parameters are optional and can be either overridden with execution-time parameters or defaults can be used if not specified by either of these methods.

**number**

Specifies the number of VTAM RECEIVE ANY buffers to be present in the IMS system. Valid values are from 1 to 500. The default is 16 if not specified during system definition or with execution-time parameters. You can override this specification with the EXEC parameter RECA= in the IMS procedure.

|  
|  
|

When MNPS is used for XRF, this parameter specifies the number of buffers for the MNPS ACB. IMS automatically allocates one additional buffer for the APPLID ACB.

Specifying too many buffers does not improve performance and can result in excessive resource consumption, especially high CPU utilization. You can check current and maximum use by looking in the X'450D' log record.

**size**

Specifies the size of the largest RECEIVE ANY buffer. This size must be large enough to handle maximum input from any VTAM-attached terminal. The minimum acceptable size is 50; the maximum is 30720. The default is 1920 if not specified during system definition or with execution-time parameters. The resulting usable input buffer size for any terminal can be calculated as follows for the devices shown:

**3601:** Size is record size plus header. The header size is 2 bytes if MFS is not used and 11 bytes if MFS is used.

**3270:** Size is the input data stream length. The minimum size is 300 bytes and a reasonable maximum size is 3842 bytes.

**SLU 1, SLU 2:** Record size should be equal to or greater than the largest input record to be received.

**SLU P:** Size is record size plus header. The header size is 5 bytes if MFS is not used and variable from 7 to 40 bytes if MFS is used.

**LU 6.1:** Size is record size plus header. The header size without MFS is variable from 0 to 45 bytes. The header size with MFS is variable from 17 to 52 bytes.

**NTO:** Record size should be equal to or greater than the largest input record to be received.

For all VTAM terminals, the RECANY buffer size must be a decimal value that can be expressed by the algorithm  $X \text{ times } 2 \text{ to the power of } Y$ . X must be a value from 8 through 15, and Y must be a value from 3 through 11. This is a VTAM restriction. Thus, for example, the value 144 (representing  $9 \times 2^4$ ) or the value 28672 (representing  $14 \times 2^{11}$ ) are acceptable values.

**Related Reading:** Details of this VTAM restriction are in *ACF/VTAM Macro Language Reference*.

This restricts the valid VTAM buffer sizes to one of the following decimal values:

112	480	2048	9216
120	512	2304	10240
128	576	2560	11264
144	640	2816	12288
160	704	3072	13312
176	768	3328	14336
192	832	3584	15360
208	896	3840	16384
224	960	4096	18432
240	1024	4608	20480
256	1152	5120	22528
288	1280	5632	24576
320	1408	6144	26624
352	1536	6656	28672
384	1664	7168	30720
416	1792	7680	
448	1920	8192	

IMS converts the specified decimal value to the format required for the bind parameter fields and places the value into the appropriate field in the bind parameter list. If the maximum request unit (RU) value specified cannot be converted exactly into the bind format, the value is rounded down to the next-lower value bind format for inbound RUs and rounded up to the next-higher value bind format for outbound RUs.

**SECCNT=**

Specifies the maximum number of terminal and password security violations to be accepted per physical terminal prior to master terminal notification of such violation. The default is 0, which nullifies notification of the master terminal. The number specified can be 0, 1, 2, or 3. This value is reset to 0 upon successful signon and does not continue across transactions.

If SECCNT is not 0, the master terminal is notified for every violation.

**SIMEXIT=**

Specifies whether (YES) or not (NO) the Shared Printer Message Router exit routine is to be included in the IMS nucleus.

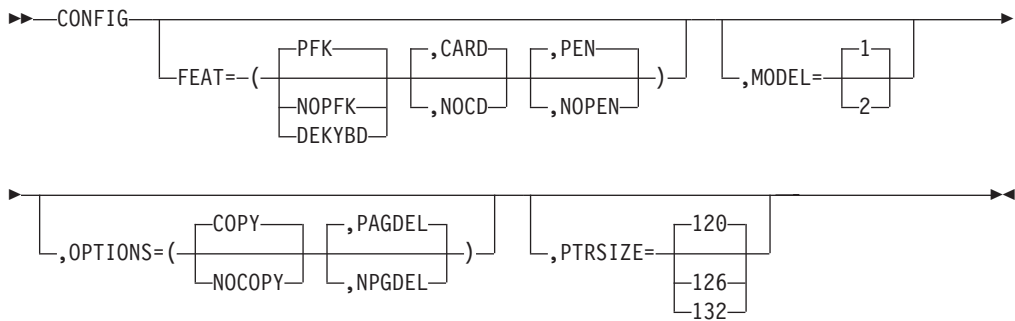
This exit routine must be placed in the data set defined by the USERLIB= keyword of the system definition IMSGEN macro statement prior to the execution of stage 2. The name of the exit routine must be DFSSIML0. A default exit routine is not provided.

The exit routine is bypassed unless SIMEXIT=YES is specified. If SIMEXIT=YES is specified, the exit routine DFSSIML0 is called when a message is queued for terminals defined with OPTIONS=SHARE on the TERMINAL macro statement.

When SIMEXIT=NO, the default, the router module assumes that /OPN is always simulated when output is enqueued for a terminal defined as OPTIONS=SHARE on the TERMINAL macro statement. Even if SIMEXIT=NO is specified, the module DFSSIML0 can be manually bound into the nucleus.

**CONFIG Macro**

The CONFIG macro statement provides the configuration for a switched 3275 terminal. Because the configuration provided by CONFIG is referenced when the named 3275 dials into IMS, differently configured 3275s can use the same communication line. All CONFIG macro statements must be between the LINEGRP macro and the LINE macros. LINE macros can refer to named CONFIG macros defined previously in this line group or in previously defined line groups.





## Label Field

Specifies a one- to eight-character alphanumeric name. This CONFIG name can be referenced in subsequent LINE and IDLIST macro instructions.

## Keyword Parameters

To find which parameters apply to your IMS configuration, refer to Table 15 on page 33.

### FEAT=

Describes this 3275:

- The first parameter specifies whether program function keys (PFK), no program function keys (NOPFK), or a data-entry keyboard (DEKYBD) is available. The default is PFK.
- The second parameter specifies whether (CARD) or not (NOCD) the operator identification card reader is available. The default is CARD.
- The third parameter specifies whether (PEN) or not (NOPEN) the selector light pen is available. The default is PEN.

### MODEL=

Specifies the terminal model number. The default is 1.

### OPTIONS=

Specifies operational options for this 3275:

- The first parameter specifies whether (COPY) or not (NOCOPY) the copy function is requested. The default is COPY.
- The second parameter specifies whether (PAGDEL) or not (NPGDEL) the automatic page deletion function is requested. The default is PAGDEL.

### PTRSIZE=

Specifies the platen size of the attached 3284 Model 3. The default is 120.

**Example:** The following example provides the configuration for a switched 3275 Model 2 with a printer platen size equal to 120. The features available are data entry keyboard and selector light pen. The copy and automatic page deletion functions are requested.

```
C1 CONFIG MODEL=2,FEAT=(DEKYBD,NOCD)
```

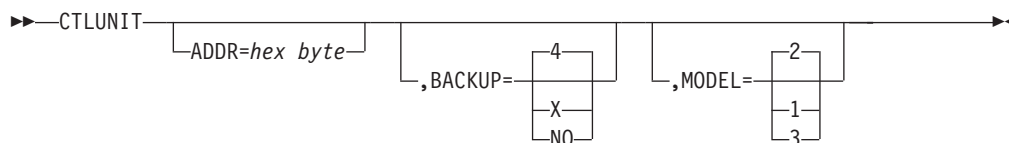
---

## CTLUNIT Macro

The CTLUNIT macro statement specifies 2848, 2972, and 3271 control unit characteristics.

- For 2848 control units, CTLUNIT specifies the control unit address, write line addressing (WLA) feature, and 2848 model number.
- For 2972 control units, CTLUNIT specifies control unit address and buffer size.
- For 3271 control units, CTLUNIT specifies control unit address and model number.

CTLUNIT is valid only for 3270 remote line groups.



## Keyword Parameters

To find which parameters apply to your IMS configuration, refer to Table 15 on page 33.

### ADDR=

Specifies the two-digit hexadecimal address of the 3271 associated with the remote 3277, or the two-digit hexadecimal address of the 2972 control unit. The address was assigned by the IBM customer engineer upon installation of the control unit. For the 3271, the IBM customer engineer assigns the selection address, which must be converted to the polling address for specification here. To be acceptable as a valid 2972 control unit address, this operand must be the hexadecimal representation of a character from A through Z.

### BACKUP=

Specifies (only for XRF) the control of the automatic restart after takeover; it is used only when HSB=YES is specified in the IMSCTRL macro.

X is a numeric integer from 1 (lowest) to 7 (highest), inclusive, that specifies the priority assigned to reestablishing the session.

When either the keyword or the parameter is omitted, the default is 4. If BACKUP=NO is specified, the automatic restart of the devices is suppressed.

### MODEL=

Specifies the 2848 or 3271 control unit model number.

For 3271 control units only, the value 1 or 2 can be specified. The default is 2.

For 3274-1C control units, only the value 2 can be specified.

For 2848 control units, the entered value determines the input buffer size for terminals attached to this control unit. The default value is 3.

---

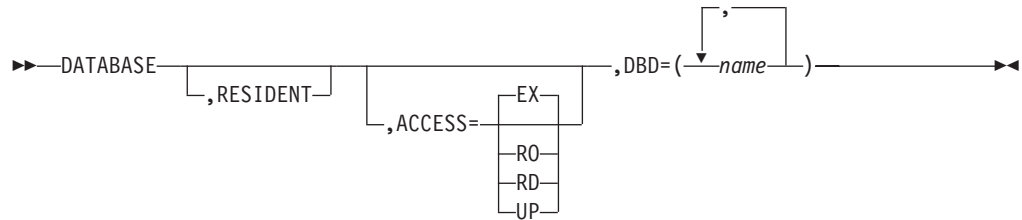
## DATABASE Macro

The DATABASE macro statement is used to define the set of physical databases that IMS is to manage. One DATABASE macro instruction must be specified for each HSAM, HISAM, and HDAM database. Two DATABASE macro instructions are required for a HIDAM database: one for the INDEX DBD and one for the HIDAM DBD. One DATABASE macro instruction must be included for each secondary index database that refers to any database defined to the online system.

For Fast Path, a DATABASE macro statement must be included for each Main Storage Database (MSDB) and Data Entry Database (DEDB) to be processed.

A database that is defined in an ALL or MODBLKS system definition cannot be converted into a HALDB partition without a cold start of IMS. A cold start is required even if the database is deleted online. After the cold start, the database must be redefined as a HALDB partition.

| A database that is defined as a HALDB partition to DBRC and IMS cannot be  
 | redefined in an ALL or MODBLKS system definition without a cold start of IMS. A  
 | cold start is required even if the database is deleted online.



## Positional Parameter

### RESIDENT=

Indicates that the DMB directory control block created for this DATABASE statement should reside in storage during system and DBCTL initialization.

Fast Path makes the database control blocks resident in storage during system initialization regardless of whether the RESIDENT option is specified.

Online changes to the RESIDENT option are permitted. However, DMBs associated with new databases defined as RESIDENT are not made resident until the next IMS restart. Until that time, they are treated as nonresident. Changes to a RESIDENT DMB (from an ACBGEN) cause that DMB to be treated as nonresident until the next restart of IMS.

## Keyword Parameters

### ACCESS=

When used in conjunction with the database-sharing level declared to database recovery control (DBRC), this parameter specifies the access for the defined database, that is, how the subsystem requesting access plans to use the database.

The ACCESS parameter can be modified by a /START command during IMS execution. Because this parameter can be changed using a /START DATABASE command, online changes to the ACCESS keyword have no effect until the next cold start.

You cannot specify an ACCESS value for MSDBs and GSAM. The possible ACCESS values are:

#### EX (exclusive)

Indicates that the named database can be accessed exclusively by this IMS subsystem. This is the default for ACCESS. Concurrent access to this database by other subsystems is prohibited by DBRC. Unless this is a DEDB, PCBs that refer to this database are scheduled, regardless of the PROCOPT values specified in PSBGN. If this is a DEDB, PROCOPT=EX is not permitted.

Concurrent image copy cannot be run against a database with ACCESS=EX. If you are running with high-speed sequential processing (HSSP), you must specify ACCESS=EX or ACCESS=UP.

**RD (read)**

Indicates that the named database can only be read by this subsystem. The database is opened for input only. No application programs are allowed to physically update this database.

**RO (read only)**

Indicates that the named database is read-only in this IMS subsystem. This database is opened for input only. PCBs that refer to this database are allowed only if PROCOPT=GO is specified in PSBGEN.

**UP (update)**

Indicates that the named database can be updated and read. PCBs that refer to this database can be scheduled regardless of their PROCOPT values.

If you are running with high-speed sequential processing (HSSP), you must specify ACCESS=UP or ACCESS=EX.

**Related Reading:** To understand the interaction between the ACCESS parameter on the DATABASE macro and the DBRC and IRLMNM parameters on the IMSCTRL macro statement, see *IMS Version 9: Administration Guide: System*.

**DBD=**

Specifies the name of one or more database descriptions (DBDs). If more than one DBD name is specified, each is assumed to have the same characteristics as any positional parameter specified or used by default. The first character of the name must be a valid alphabetic character (A through Z, #, \$, or @). At execution time, the DBD must have been processed by the block builder utility program. It must exist as a member in the partitioned data set named in the IMSACB DD statement (IMS.ACBLIB). If the DBD has not been processed by the block builder utility program, and is, therefore, not present in the ACBLIB data set, the database is locked at execution time. This parameter is required.

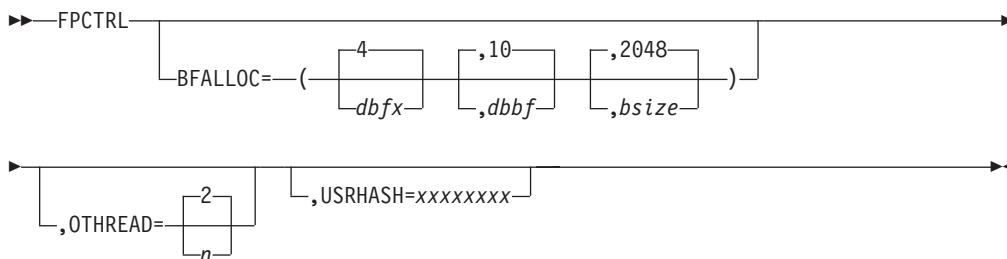
**Example:**

DATABASE DBD=(N1,N2,N3)

---

**FPCTRL Macro**

The FPCTRL macro statement defines the IMS Fast Path options of the IMS control program, and the DBCTL environment. This statement must appear somewhere between the IMSCTRL and MSGEN macro statements if Fast Path is used. It is ignored when the IMSCTRL statement specifies that only a BATCH or MSVERIFY system definition is to be performed. If Fast Path is to be included in a DCCTL environment, specify the FPCTRL macro without any parameters.



## Keyword Parameters

To find which parameters apply to your IMS configuration, refer to Table 15 on page 33.

**Related Reading:** If Fast Path is implemented in your IMS configuration, see “Member DFSFIXnn” on page 352 for Fast Path page fix recommendations.

### **BFALLOC=**

Specifies the values used to allocate a storage pool from which storage for control blocks and buffers associated with the Data Entry Database (DEDB) and Main Storage Database (MSDB) is obtained. The default values are used unless these are overridden by EXEC statement parameters when an IMS online control region is executed. See “Specifying IMS Execution Parameters” on page 415.

For an RSR tracking subsystem, these database buffers are used as I/O staging buffers. Also for RSR, be sure to convert your MSDBs to DEDB areas or DEDB VSO areas.

**Related Reading:** See *IMS Version 9: Utilities Reference: Database and Transaction Manager*.

### **dbfx**

Specifies additional buffers (not separate pools) to be page-fixed at the start of the first Fast Path region. This specification allows for asynchronous processing where the DEDB updates are held until the associated log buffer is written. The default is 4.

This parameter is not used by an RSR tracking subsystem.

### **dbbf**

Specifies the total number of database buffers. The specification should satisfy the following formula:

*dbbf* must be equal to or greater than the sum of open DEDB areas that have sequential dependent segments defined, plus:

- The sum of all Fast Path buffers used by CCTLs (CNBA)
- The sum of all database buffers (NBAs) for any executing Fast Path-dependent region
- The *dbfx* specification
- The maximum overflow buffer allowance (OBA) for any executing Fast Path region

The valid range is 1 to 65535. If you intend to use a DEDB or MSDB, 0 is not reasonable. Insufficient allocation of buffers can result in an open error for an area or in a start error for a region. The default is 10.

### **bsize**

Specifies the size of the database buffers. It must be equal to or greater than the largest control interval size used by the DEDB. The only valid values are 512, 1K, 2K, 4K bytes, and multiples of 4K bytes up to 28K bytes. The default is 2048 bytes (2K bytes).

### **INREXIT=**

Is no longer used. An exit routine DBFHAGU0 is automatically included for Fast Path. If you were using the DBFSUIX0 sample exit routine, the new sample exit routine DBFHAGU0 should be compatible.

**Related Reading:** See the *IMS Version 9: Customization Guide* for more information.

#### OTHRD=

Specifies the number of concurrent output threads that IMS Fast Path is to permanently allocate. For an RSR tracking subsystem, the OTHR= parameter is used to determine the maximum number of concurrent I/O operations for Fast Path database tracking.

The valid range is from 1 to 255, with a default of 2. Each output thread is dispatched in the control region and performs the output functions for the Data Entry Database (DEDB).

#### USRHASH=

If this parameter is specified in the FPCTRL macro, it will be ignored.

**Example:** The following is an example of the FPCTRL macro statement.

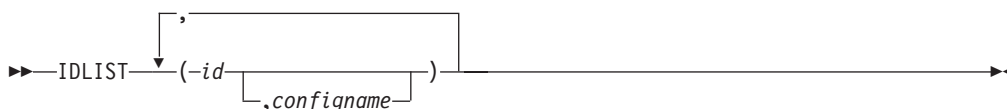
```
FPCTRL OTHR=3,BFALLOC=(30,100,1024)
```

## IDLIST Macro

The IDLIST macro statement is used to create a terminal security list for switched 3275s. Each entry consists of a 3-byte hexadecimal value (derived from the last 3 bytes of the terminal's 4-byte hardware ID) and the name of a previously defined configuration. The maximum number of ID entries—(*id,configname*) is an entry—that can be specified in an IMS system definition is 5000. If the IDLIST macro is used, it must follow the LINE macro.

If the IDLIST macro is not used, the list specified by IDS= in the preceding LINE macro is used to determine the 3275 configuration. If IDS= is not specified, the terminal configuration used is the same for all terminals that dial in on this line (either the CONFIG= specified in the LINE macro or the IMS default).

At least one IDLIST macro is required if VERIFY=YES is specified on any LINE macro. A LINE macro that specifies VERIFY=YES must either also specify IDS= or be followed by an IDLIST macro.



## Label

A one- to eight-byte alphanumeric name, which the IDS= operand of a subsequent LINE macro can reference.

## Positional Parameters

To find which parameters apply to your IMS configuration, refer to Table 15 on page 33.

**id** Specifies in hexadecimal the last three bytes of the terminal's 4-byte hardware ID.

#### configname

Specifies the name of the configuration to be used for this terminal. If *configname* is specified, the name provided must reference a previously defined CONFIG macro. If *configname* is not specified, the configuration named in the

CONFIG= operand of the associated LINE macro is used. If no configuration is specified in the LINE macro, the IMS default is used.

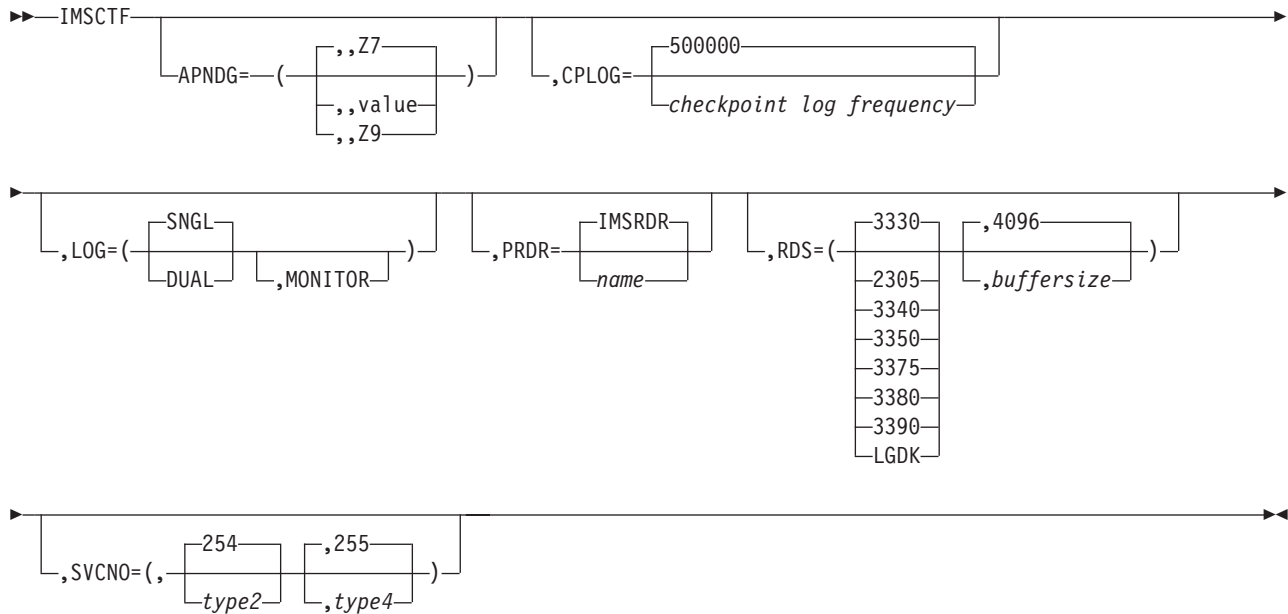
**Example:** The following is an example of the IDLIST for four switched 3275s:

```
IDLIST (E54C5A,A),(E54C5B),(E54C5C,C2),(E54C5D)
```

The terminal IDs are E54C5A, E54C5B, E54C5C, and E54C5D. Terminal E54C5A uses the configuration described by the CONFIG macro named A. Terminal E54C5C uses the configuration described by the CONFIG macro named C2. Terminals E54C5B and E54C5D use either the configuration named in the LINE macro or the IMS default configuration.

## IMSCTF Macro

The IMSCTF macro statement defines parameters to IMS, and to the DBCTL environment. The DYLOG and DISKLOG keywords and the third parameter (2|number) on the RDS keyword are no longer used and are not included in the macro format described in this section. If either of these are specified, they are ignored. They can remain in your system definition to provide compatibility with previous IMS releases.



## Keyword Parameters

To find which parameters apply to your IMS configuration, refer to Table 15 on page 33.

### APNDG=

The first operand and second operands are no longer used. The comma is retained to maintain compatibility with previous releases.

The third operand specifies the Multiple Systems Coupling channel-to-channel appendage suffix for z/OS systems. The value specified can range from WA to Z9 (the default is Z7).

User appendages IGG019Z7 or IGG019Z8 can already exist. If any of these appendages do exist, you should do one of the following:

- Rename the existing appendages
- Do not permit the APNDG parameter to default to Z7, but instead choose other suffixes

**CORE=**

No longer used. CORE= has been replaced by the PIINCR and PIMAX execution parameters for DB/DC and DBCTL IMS online systems, respectively.

**CPLOG=**

Specifies the number of system log records between system-generated checkpoints. The permitted value ranges from 500 to 16777215. The default is 500000.

**LOG=**

The first operand, SNGL|DUAL, is no longer used for an online execution. If specified in an online configuration, it is ignored.

When specified for a batch execution, this parameter determines whether 1 (SNGL) or 2 (DUAL) DD statements are to be generated in the IMS batch procedures for the system log.

The second operand, MONITOR, specifies that the IMS Monitor DD statement is to be included in the procedure for the online system.

**PRDR=**

Specifies an optional IMS reader proc name. The name must be one to eight-alphanumeric or national characters, with the first being alphabetic or national. The default is IMSRDR. You can also specify the reader proc name in the control region JCL and in the default parameter module, DFSPBIMS. DBCTL users use DFSPBDBC.

**RDS=**

The first parameter specifies the device type on which the restart data set (IMS.RDS) is to reside.

You can specify one of the following device types: 3330 (the default), 2305, 3340, 3350, 3375, 3380, 3390 or LGDK.

When using a 3350, specify the drive format used—3330 or 3350.

LGDK is a generic definition for disk drives that have a track size greater than 32767 bytes. The 3375, 3380, 3390, and future devices can also be specified as LGDK.

The second parameter specifies the buffer size to be used for the data set. The minimum buffer size is 1024. The maximum allowable size is the track size for the device type specified or 32767, whichever is smaller. The default is 4096.

The third parameter is no longer used. If specified, it is ignored.

**SVCNO=**

Specifies supervisor call instruction (SVC) numbers reserved for use by the generated system.

The first parameter specifies the type 2 SVC number. The entered value can range from 200 to 255. The default is 254. The type 2 SVC number is required for batch, DBCTL, DCCTL and DB/DC IMS control program functions. One type 2 SVC number can be used for any number of IMS systems.

The second parameter specifies the type 4 SVC number reserved for use by database recovery control (DBRC). The entered value can range from 200 to 255. The default is 255. One type 4 SVC number can be used for any number of IMS systems.



For reasons of compatibility, the SVC numbers, if specified, must be immediately preceded by a comma and then enclosed in parentheses. Also, the type 2 and type 4 SVC number cannot be the same.

|  
|  
|

If you are installing different levels of IMS in the same processor, note that the Type 2 and Type 4 SVCs are downward compatible. The Version 9 level can be used by Version 8. However, Version 8 cannot be used by Version 9.

**Related Reading:** See “OS/390 Interface Considerations” in the *IMS Version 9: Installation Volume 1: Installation Verification* for compatibility information regarding the SVC numbers used for different versions of IMS.

**Example:** Following is an example of the IMSCTF macro instruction for the following condition:

- Type 2 SVC is 254.
- DBRC Type 4 SVC is 255.
- Checkpoint frequency is every 2000 log records.
- The maximum dynamic storage allowed is 16K.
- The increment is 4K.

```
IMSCTF SVCNO=(,254,255),APNDG=(,Z0),CPLDG=2000
```

## IMSCTRL Macro

The IMSCTRL macro statement describes the basic IMS control program options, the z/OS system configuration under which IMS is to execute, and the type of IMS system definition to be performed.

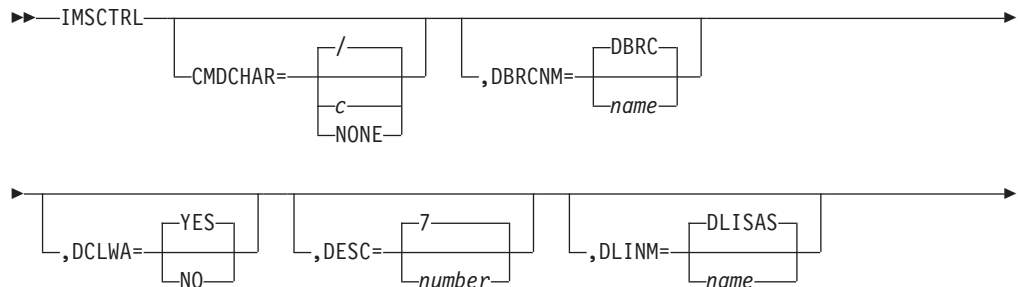
IMSCTRL is required. This macro instruction must be the first statement of the system definition control statements that are entered into the system definition Stage 1 for an IMS or a DBCTL system. You can have only one IMSCTRL macro instruction within an IMS system definition.

The only other macro instruction required for a batch environment is IMSGEN. The IMSCTF macro instruction is optional.

|  
|  
|  
|

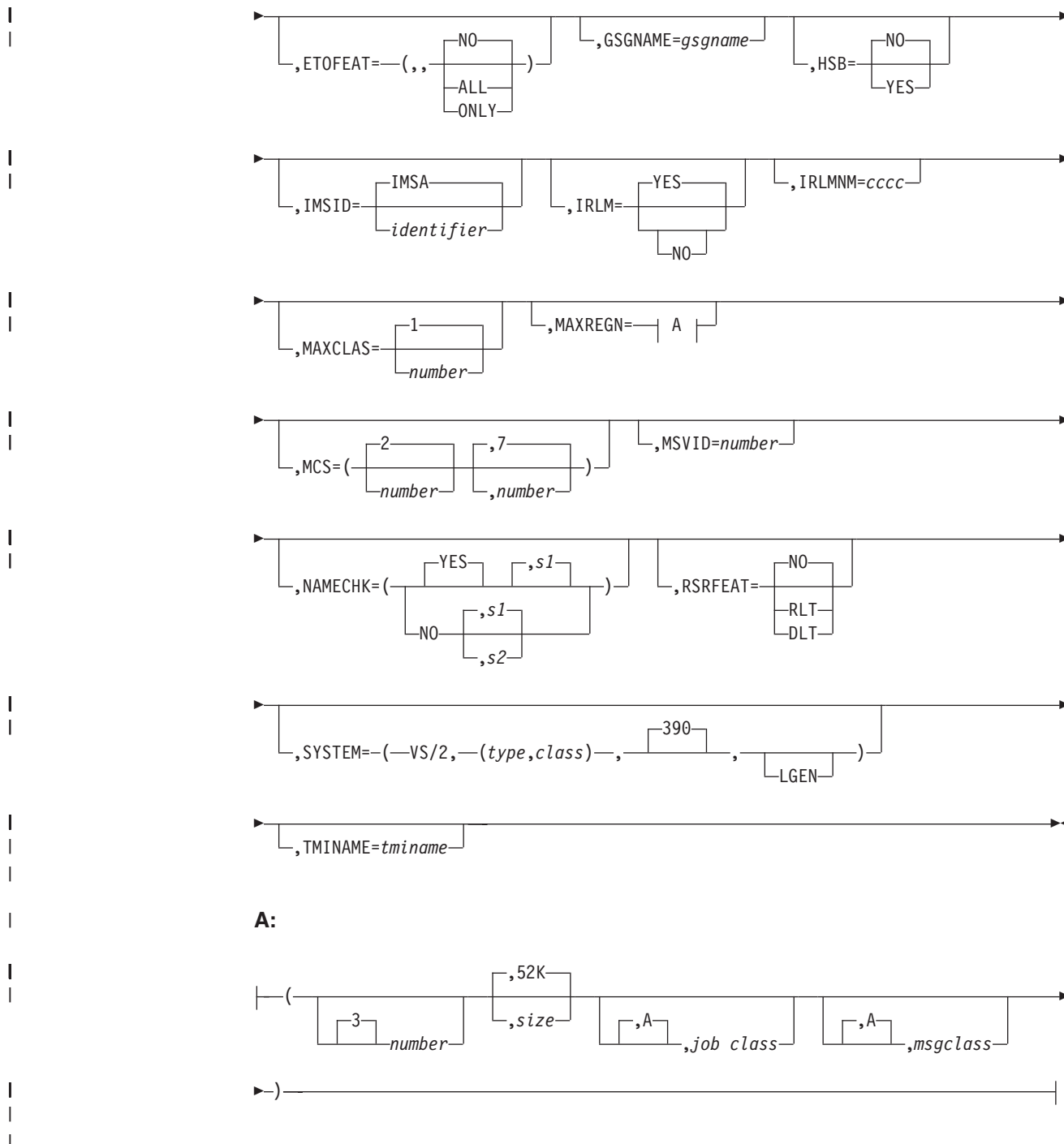
**Note:** If you run stage 1 and stage 2 processing using the IMSCTRL macro without specifying MODBLKS, a cold start is required. Failing to specify MODBLKS causes updates to IMS.SDFSRESL, which requires the new IMS.SDFSRESL to be copied and, therefore, a cold start is necessary.

|  
|



|  
|

|



**A:**

**Keyword Parameters**

To find which parameters apply to your IMS configuration, refer to Table 15 on page 33.

**CMDCHAR=**

- / The default value.
- c Any character other than comma (,). Ampersand must be entered as && to define & as the command character.

**NONE** No command recognition character (CRC) is defined.

CMDCHAR can be specified for DBCTL, DB/DC or DCCTL. The value of CMDCHAR can be specified at execution time by specifying execution parameter CRC=c.

With a DBCTL system, CMDCHAR specifies the command recognition character used by the MTO to enter IMS commands for this DBCTL environment. If CMDCHAR=NONE is specified and the CRC is not specified as an execution parameter (CRC=), then commands are entered for this DBCTL system with a prefix of the IMSID.

DB/DC and DCCTL systems allow commands to be entered at MCS/E-MCS consoles with a CRC prefix by specifying execution parameter keyword CMDMCS with a value other than N.

The CRC can be specified with CMDCHAR or with CRC at execution time.

If CMDCHAR= is specified and the CRC is not specified as an execution parameter (CRC=), commands are entered for the DB/DC or DCCTL system with a prefix of the IMSID.

Although other products and subsystems might use certain command recognition characters, the CRC is not required to be unique among all IMS systems. Choose an unreserved character that is appropriate for your particular system.

Do not use a character for the CRC that is the same as the beginning character of an z/OS command. If the CRC is the same as the beginning character of an z/OS command, that command does not work after you start IMS. For example, if you start IMS with CRC=D, z/OS does not respond to any of the z/OS display commands such as D A,L.

The CRC value you select might affect multisegment command processing. The last character of the command is checked during multisegment command processing. If that character matches the CRC, the command is assumed to be a multisegment command. Processing of the command waits for the rest of the command. For example, if you use B for the command recognition character, and issue the command 'BMODIFY PREPARE ACBLIB', the command does not run. The ending 'B' in 'ACBLIB' is interpreted to indicate a multisegment command. To enter this command correctly, enter:

```
'BMODIFY PREPARE ACBLIB. '
```

The character you choose for a DB/DC system becomes the command recognition for the IMS system. If your installation allows commands from MCS or E-MCS consoles, you need a CRC only if both of the following statements are true:

- IMS provides DBCTL service to a CCTL.
- CCTL broadcasts an IMS SWITCH command preceded by a CRC.

CMDCHAR does not affect the normal DB/DC MTO's use of the / character when entering IMS commands.

#### **DBRC=**

The system definition process ignores this parameter, though for compatibility purposes, it can still be coded. See "DBRC" on page 420 for information about changing the current value for this parameter.

#### **DBRCNM=**

Specifies for z/OS the IMS.PROCLIB member containing the skeletal procedure for the DBRC address space, produced by IMS system definition. The IMS control region automatically initiates the DBRC address space, using the z/OS

start command and specifying this name. To rebuild the DBRC procedure, you must use the IMS.PROCLIB member containing the skeletal procedure for the DBRC address space produced during system definition. This procedure contains the name you specify in stage 1. You must first change DFSPBIMS, DFSPBDBC, or DFSPBDCC to reflect the specified DBRCNM, and then perform another system definition.

Consequently, the DBRC procedure produced by IMS system definition must be copied from IMS.PROCLIB to SYS1.PROCLIB. For more information, see “Making IMS and IMSRDR Procedures Accessible to z/OS” on page 433.

You can specify up to an eight-character name, the first character being alphabetic. The default name is DBRC.

Specifying DBRCNM=xxxxxxx results in the following:

A DBRC address space cataloged procedure is created with the name xxxxxxxx.

To make the DBRC address space cataloged procedure name of xxxxxxxx operational, the following must be true:

A DBRC address space must be cataloged as xxxxxxxx in SYS1.PROCLIB.

Simply specifying DBRCNM=xxxxxxx in your IMS system definition is not sufficient.

#### **DCLWA=**

Establishes a default value that IMS applies to any transactions defined by subsequent TRANSACT macro statements. DCLWA= specifies whether (YES) or not (NO) IMS should perform log write-ahead for recoverable nonresponse-mode input messages and transaction output messages. The default is YES. If no value is specified for DCLWA= on the IMSCTRL macro, the value for DCLWA= on any subsequent TRANSACT macros accepts the default of YES, or you can specify YES or NO on the individual TRANSACT macros.

Specify or accept the default of YES to ensure that:

- A nonresponse–input transaction is made recoverable across IMS failures, prior to IMS acknowledging receipt of the input.
- Database changes are made recoverable prior to IMS sending associated output reply messages.

YES ensures that information in the log buffers is written to the IMS log before the associated input acknowledgment or output reply is sent to the terminal.

Specify or accept the default of YES for all VTAM terminal types. For BTAM terminals, selection of this option can increase input and output message path lengths.

**Related Reading:** For information about performance implications of selecting log write-ahead for transactions, see *IMS Version 9: Administration Guide: System*.

Specify NO if input message integrity and the consistency of output messages with associated database updates is not required. DCLWA does not apply to response mode or Fast Path input processing, and, if specified in these situations, is ignored during IMS execution.

#### **DESC=**

Specifies the message descriptor codes to be assigned to the IMS system

console messages if MCS support is included in the z/OS system definition. You can specify up to 16 values. If DESC is not specified, the default value is 7.

The IMS MCS= and DESC= keywords should be defined as is required for the ROUTCDE and DESC keywords of the z/OS WTO macro.

#### **DLINM=**

Specifies for z/OS the IMS.PROCLIB member to contain the skeletal procedure for the optional DL/I address space produced by IMS system definition. The IMS control region automatically initiates the DL/I address space using the z/OS start command and specifying this name.

Consequently, the DL/I procedure produced by IMS system definition must be copied from IMS.PROCLIB to SYS1.PROCLIB. You probably need to modify the skeletal procedure.

For more information, refer to “Member Name DLISAS” in Chapter 4, “Procedures,” on page 191.

You can specify up to an eight-character name, the first character being alphabetic. The default name is DLISAS.

Specifying DLINM=xxxxxxx results in the following:

A DL/I address space cataloged procedure is created with the name xxxxxxxx.

#### **ETOFEAT=**

Specifies whether or not the system definition process should build ETO descriptors. The default is not to build the ETO descriptors.

The first two parameters of the ETOFEAT keyword are no longer valid and will be ignored if specified. Commas appear in place of these parameters in the syntax diagram.

The third parameter indicates whether the system definition process should produce ETO descriptors only (ONLY), produce descriptors along with a normal system definition (ALL), or produce a normal system definition without descriptors (NO). The default is NO.

If you have not installed ETO, the ETO option is forced off.

#### **GSGNAME=**

Specifies the global service group (GSG) name to be used for the RSR complex. No default value exists. If you have the RSR feature (RLT, DLT, or both) installed and you specify a GSG name, RSR is enabled for this system definition.

If you do not specify the GSG name in the IMSCTRL macro and you want RSR enabled, the GSG name must be specified in the DFSRSRxx member (or the JCL for batch or utility jobs) to enable RSR. If you do specify the GSG name in the IMSCTRL macro, you might have to change batch JCL to allow batch jobs to run with RSR not enabled.

**Recommendation:** If RSR is not installed or you do not want RSR enabled for this system definition, IBM recommends that you do **not** specify a value for GSGNAME.

If you do specify a value for GSGNAME and you do not want RSR enabled, you must override the RSR enablement by specifying RSR(NO) in the DFSRSRxx PROCLIB member. For more information, see “Member DFSRSRxx” on page 366.

**HSB=**

Specifies whether (YES) or not (NO) an XRF-capable system is to be generated. The default is NO.

Specifying NO prevents the possibility of an XRF capability without rerunning the system definition and an IMS cold start.

**Recommendation:** If you are planning on running XRF, code HSB=YES on the IMSCTRL macro. Having HSB=YES coded might eliminate a subsequent IMS system definition.

XRF is disabled if no value is specified for HSBID in the DFSPBIMS or DFSPBDCC members, or in the IMS execution JCL. Informational message DFS3899I is issued.

**IMSID=**

Specifies a one- to four-character alphanumeric identifier for the IMS system.

The IMSID value specified is used as the IMS subsystem identifier. For online control regions, it must be different from any other IMS subsystem identifier or non-IMS subsystem identifier defined to the operating system under which IMS is running. This identifier is also used to relate messages that are routed to the z/OS system console with the corresponding IMS system. To avoid confusion as to which region has issued a console message, it is recommended (though not required) that the IMSID be unique for batch regions. The default is IMSA.

Do not use characters for the IMSID that match the beginning characters of an z/OS command. If the IMSID is the same as the beginning characters of an z/OS command, that command does not work after you start IMS. For example, if you start IMS with IMSID=D, z/OS does not respond to any of the z/OS display commands such as D A,L.

If you have renamed the IMS procedure (the procedure that starts IMS), the IMSID value specified must not be the same as the IMS procedure name unless one of these conditions exists:

- All the DD statements in the startup procedure are cataloged in the master catalog
- The unit and volume are specified on each DD statement

**IRLM=**

If coded as IRLM=NO, sets the IRLM batch default to NO. This lets you generate IMS with IRLM active for the online system, but inactive for the batch environment. The NO can be overridden by the IRLM=Y parameter in the batch JCL.

If the IRLM= parameter is omitted, and the IRLM name is specified, the IRLM batch default is set to YES, but can be overridden by the IRLM=N parameter in the batch JCL.

If IRLMNM is not coded, this parameter is ignored. (See the IRLMNM parameter.)

To avoid assembly errors, including 2,G006, you must specify the combination of the IRLM and IRLMNM parameters in one of the following ways:

- If you specify IRLM:
  - IRLM=Y and IRLMNM=xxxx
  - omit IRLM and specify IRLMNM=xxx
- If you omit IRLM:
  - IRLM=N and IRLMNM=xxxx.

- omit **both** parameters

If you specify the IRLM= parameter and omit the IRLMNM= parameter, assembly errors can result.

#### **IRLMNM=**

Specifies a one- to four-character alphanumeric name assigned to the Internal Resource Lock Manager (IRLM) included in your system.

If the IRLMNM= parameter is not specified, the IRLM is not included in your IMS system. To activate the IRLM, you must code IRLM=Y in the execution JCL.

When the IRLMNM= is not specified in the IMSCTRL macro or in the execution JCL, and IRLM=Y is specified in the execution JCL, the IRLM name used is IRLM.

The IRLM must be active in your system if your IMS system is to share data at the block level with other IMS systems. The IRLM is optional for systems that do not share data at the block level.

Only one IRLM can be specified for any IMS system (and thus, only one IRLMNM), but multiple IRLMs can exist within a particular z/OS system. Each IRLM must be defined as an z/OS subsystem.

**Related Reading:** For more information, refer to *IMS Version 9: Installation Volume 1: Installation Verification*.

The name specified on this IRLMNM= parameter must be the z/OS subsystem name assigned to the IRLM with which this IMS system is to connect. If IMS is connected to an executing copy of IRLM, DB2 cannot connect to it. If DB2 is connected to an executing copy of IRLM, IMS cannot connect to it.

If this parameter is specified and not overridden by means of the IRLM=NO parameter on the online or batch startup procedures, IMS only completes initialization if the named IRLM is available at execution time. Also, IMS uses the named IRLM for all locking services, both local and sharing. Program isolation locking is not used.

It is important to understand the interaction of this IRLMNM parameter with the DBRC parameter and with the ACCESS parameter on the DATABASE macro statement.

**Related Reading:** For additional information on administering a data-sharing environment, see *IMS Version 9: Administration Guide: System* before defining your data sharing options.

#### **MAXCLAS=**

Specifies the range of active (started) classes after IMS initialization. Valid values are decimal numbers from 1 to 999. The default is 1. Any transaction code class specified on either the APPLCTN or TRANSACT macro statement must not exceed this value. If the class is not active at the time of scheduling, the schedule request is treated as if the class were stopped.

#### **MAXIO=**

Is no longer used. If MAXIO is specified the value is ignored.

#### **MAXREGN=**

Specifies the maximum number of IMS control block sets that are permanently allocated. A control block set is required for each dependent region type (batch message processing, message processing, or CCTL threads) that can be started. (Note that this does not imply that the same number of dependent regions are always active.)

This value must be a decimal number from 1 through 999. When the count of dependent regions started exceeds the value specified by the MAXREGN= parameter, control block sets are dynamically allocated, but the storage for these sets is released as the dependent regions terminate. The default is 3.

For an RSR tracking subsystem, the value specified on the MAXREGN= parameter determines the maximum number of PSTs that are to be used for DL/I database tracking. This number should be twice the number of DL/I PSTs used in all of the active subsystems tracked by the tracking subsystem. If many PSTs are non-updating, set this number to twice the number of updating DL/I PSTs used in all active subsystems. A minimum of two PSTs is used.

The second through fourth fields specify region size (default value is 52K), region job class (default class is A), and job message class (default class is A).

Region size (which must be expressed in terms of xxK), region job class, and job message class must conform to the operating system JCL specifications. This operand is ignored for BATCH system definitions.

#### **MCS=**

Specifies the z/OS routing code to be assigned to the IMS system console if multiple console support (MCS) is included in the operating system. If MCS is not specified, the default is (2,7).

Although z/OS supports more than 16 route codes, IMS uses only route codes 1 through 16.

In a DBCTL environment, MSC= defines which consoles are to receive unsolicited DBCTL messages.

#### **MSVID=**

Specifies a one- to three-digit decimal number between 1 and 676 that is used to complete the eight-character name of the generated IMS Multiple Systems Verification utility control blocks module generated for this system (DFSMSxxx). If fewer than three decimal digits are specified, the value is right-justified and filled to the left with zeros. (For example, specifying 3 creates the name DFSMS003.)

This control blocks module is assembled and bound during IMS system definition stage 2 processing. The Multiple Systems Verification utility procedure (IMSMSV) is, as the user directs, placed into the IMS.PROCLIB data set. This keyword is required when MSVERIFY is specified in the SYSTEM keyword.

This keyword is ignored if no multiple systems definition statements are present. The keyword must be present to generate the IMS Multiple Systems Verification utility control blocks and procedure.

This keyword is ignored for DBCTL.

The specification of MSVID on the IMSCTRL macro statement is not related to the specification of SYSID on the MSNAME macro statement.

#### **NAMECHK=**

Specifies whether (YES) or not (NO) resource name checking and cross-checking are to be performed, and, if NAMECHK=NO, whether sorting is to be performed in stage 1 (S1) or stage 2 (S2).

If duplicate resource name checking and resource name cross-checking will be performed (that is, NAMECHK is specified as or uses the default of YES), sorting must be performed in stage 1 (S1 specified or used by default). If resource name checking is to be bypassed, sorting can occur in either stage 1 (S1 specified or by default) or stage 2 (S2 specified).



It is suggested that you use the default values for NAMECHK unless:

- You have run the preprocessor prior to your stage 1 run and you want to omit resource name checking here (by specifying NAMECHK=NO).
- You have a specific performance reason for having IMS sort in stage 2, (for example, you generally run stage 1 multiple times per stage 2).

**Related Reading:** For information about the options available to you by including or omitting the preprocessor in combination with options on the NAMECHK parameter, see “Verifying the Stage 1 Input” in *IMS Version 9: Administration Guide: System*.

This keyword is ignored if LGEN is specified in the SYSTEM keyword.

#### **RSRFEAT=**

Specifies RLT or DLT for RSR. This keyword is ignored by IMS, but is left in for compatibility with prior releases. For information about enabling RSR, see GSGNAME=.

#### **SYSTEM=**

Specifies the operating system configuration and the type of IMS system definition to be performed.

The first SYSTEM= parameter must be VS/2, indicating the operating system is z/OS.

The second SYSTEM= parameter consists of two subparameters. The first subparameter defines *type* of system definition. The definition types are: ALL, BATCH, CTLBLKS, MODBLKS, MSVERIFY, NUCLEUS, and ON-LINE.

When you first install IMS, you must perform a full system definition. Thereafter, you can perform a subset of the full system definition to add or change a function. The parameters used in these subset system definitions are described in this section. The type of definition required to add or change a function is described in “Selecting the Appropriate IMS System Definition” on page 29.

The SMP JCLIN process should be run after the stage 1 of all definition types.

After an ALL, ON-LINE, NUCLEUS, CTLBLKS, or MODBLKS definition is completed, the Security Maintenance utility must be rerun.

#### **ALL**

Combines the BATCH and ON-LINE options.

#### **BATCH**

Moves required modules from the IMS distribution libraries to the user's libraries; it also generates system procedures and a database system.

A BATCH definition should not be run to add maintenance or to add or delete database functions or features to a residence library that contains a nucleus with the Data Communication feature.

#### **CTLBLKS**

Generates control block modules of all IMS control blocks for use within an IMS nucleus in the IMS control program region. The blocks built in the MODBLKS option are also included in this definition. This subparameter should be specified to replace the control blocks of an existing nucleus. If you are performing a CTLBLKS definition, the new nucleus must have the same suffix as the old nucleus. SMP maintenance integrity is lost if you rename the nucleus prior to using it in a CTLBLKS definition because SMP does not recognize the contents of the renamed nucleus. To change the

control blocks in an existing nucleus, do a CTLBLKS definition. To generate a new nucleus, do a NUCLEUS definition.

The SMP JCLIN process should be run after the stage 1 of a CTLBLKS definition.

A CTLBLKS specification generates the MSC Verification Utility control blocks, if MSC definitions are present. When you specify CTLBLKS, you need not perform a separate system definition with MSVERIFY.

If any device or option is to be deleted during a CTLBLKS definition, unresolved references may exist when the nucleus is bound. This can also happen if logical terminal names are added or deleted before the occurrence of master or secondary master logical terminal names.

### MODBLKS

Generates the control blocks members for resources to be added or changed online. These control blocks are used by the IMS control region, the Security Maintenance utility (SMU), and the Multiple Systems Coupling Verification utility.

The MODBLKS system definition requires that the stage 1 input contain all source statements for the existing system to which the online changes are to be made, with the exception of communication specifications. If you use the Multiple Systems Coupling MSVERIFY utility, **all** source statements, including communication specifications, are required input to your stage 1 run. However, if communication specifications are omitted from a MODBLKS system definition, the following stage 1 error checking is not performed:

- Transaction names are not cross-checked against LTERM names to locate duplicates.
- For MSC users, the remote system ID (SYSID) on the TRANSACT macro statement is not checked against the remote system ID (SYSID) on the MSNAME macro statement to ensure that they are not duplicates.

Thus, omitting communication statements can result in undetected system definition errors. You must therefore consider your requirements for improved stage 1 performance against the possibility of undetected errors in your stage 1 input to determine whether you want to include or omit communication statements from your MODBLKS system definition input.

Only APPLCTN, DATABASE, TRANSACT, and RTCODE statements can be added, deleted, or modified. The restrictions pertaining to these modifications are found in the descriptions of these macro statements. IMS does not enforce these restrictions; rather, you are responsible for adhering to the directions provided there.

The effects of a MODBLKS system definition are made known to the IMS online system by means of a master terminal operator /MODIFY command sequence.

Stage 2 assembler output for a MODBLKS system definition is placed in IMS.OBJDSET. Binder output for the control blocks members affected by running an online change is placed in IMS.MODBLKS.

Certain types of changes, when made across a MODBLKS system definition, do not affect an existing resource except across a cold start. That is, the effects of prior commands are not negated by online change

procedures. For example, resources that have been modified by commands such as /START, /STOP, /ASSIGN, /MSASSIGN are not altered, nor are resources whose status has been altered because of changes in the operating configuration (for example, a resource stopped because of an error condition).

**Note:** If you run stage 1 and stage 2 processing using the IMSCTRL macro without specifying MODBLKS, a cold start is required. Failing to specify MODBLKS causes updates to IMS.SDFSRESL, which requires the new IMS.SDFSRESL to be copied and, therefore, a cold start.

### MSVERIFY

Generates a subset of the online control blocks required for execution in a multiple-IMS system configuration. MSVERIFY also generates the bind JCL and control statements required to place the Multiple Systems Verification utility into the IMS.SDFSRESL data set. MSVERIFY can also generate the JCL and control statements to place the IMSMSV procedure into the IMS.PROCLIB data set. This subparameter should be specified to generate a test system to verify, prior to attempting online execution, the consistency of definitions in an MSC configuration. This keyword causes an assembly error if the multiple systems definition statements are not present. If MSVERIFY is specified, MSVID is required.

### NUCLEUS

Generates an IMS nucleus for the IMS control program region and control block modules for all IMS control blocks. This subparameter should be specified when generating an alternate nucleus that includes new or deleted old system features, such as additional terminal support or conversational support. A NUCLEUS specification generates the MSC Verification utility control blocks if MSC definitions are present. If NUCLEUS is specified, you do not need to perform a separate system definition, with MSVERIFY specified, to obtain the blocks.

To change the control blocks in an existing nucleus, do a CTLBLKS definition. To generate a new nucleus, do a NUCLEUS definition.

The SMP JCLIN process should be run after the stage 1 of all definition types.

### ON-LINE

Moves required modules from the IMS distribution libraries to the user's libraries, generates system procedures, and generates those modules that contain the CTLBLKS and NUCLEUS options for an IMS system. It generates the assembler language and bind statements for all operating systems and VTAM-sensitive modules that are supplied in source form.

You must specify ON-LINE with the hyphen.

## Guidelines for Selecting the System Definition Type

Guidelines are provided to assist you in selecting the CTLBLKS, NUCLEUS, ON-LINE, or MODBLKS system definition type to use when modifying an existing ALL type of system definition. The guidelines should help you in determining whether modification of parameters at execution can preclude the need for a new system definition. For more information, see "Selecting the Appropriate IMS System Definition" on page 29.

**Example:** The following is an example of the IMSCTRL macro with:

- Three message regions

- Message region job class A (default)
- Job message class C
- Online and batch

```
IMSCTRL SYSTEM=(VS/2,(ALL,DB/DC),390),           X
          MAXREGN=(3,52K,,C)
```

The second SYSTEM= subparameter defines the system definition environment *class*. The three system definition environment classes are:

<b>DB/DC</b>	Builds a standards IMS system DB/DC is the default.
<b>DBCTL</b>	Builds only the DBCTL environment
<b>DCCTL</b>	DCCTL class builds only the DCCTL environment

If you do not specify DBCTL on the second subparameter, the following are always included:

- All enhanced security codes and commands
- All MFS code, commands, procedures, and default formats
- VTAM code and commands, whenever VTAM is requested

The MFS format library is a required data set (regardless of whether MFS is used), except for DBCTL systems.

The MSVERIFY,DBCTL type/class combination is invalid. An error message is generated for invalid combinations.

The third SYSTEM= parameter indicates the operating system; the default is 390 (OS/390).

The fourth SYSTEM= parameter defines the LGEN configuration. Use LGEN if a stage 1 or any of the stage 2 job steps of a system definition requires more than 4 MB of private storage for execution.

The LGEN parameter is only valid when stage 1 source statements are input to the preprocessor. The preprocessor passes a group of up to 10000 resources to the assembler for stage 1 processing. The assembler is invoked as many times as is necessary to process all appropriate resources.

**Related Reading:** For more information on the LGEN configuration and how the IMS control region is organized, refer to *IMS Version 9: Administration Guide: System*.

#### **TMINAME=**

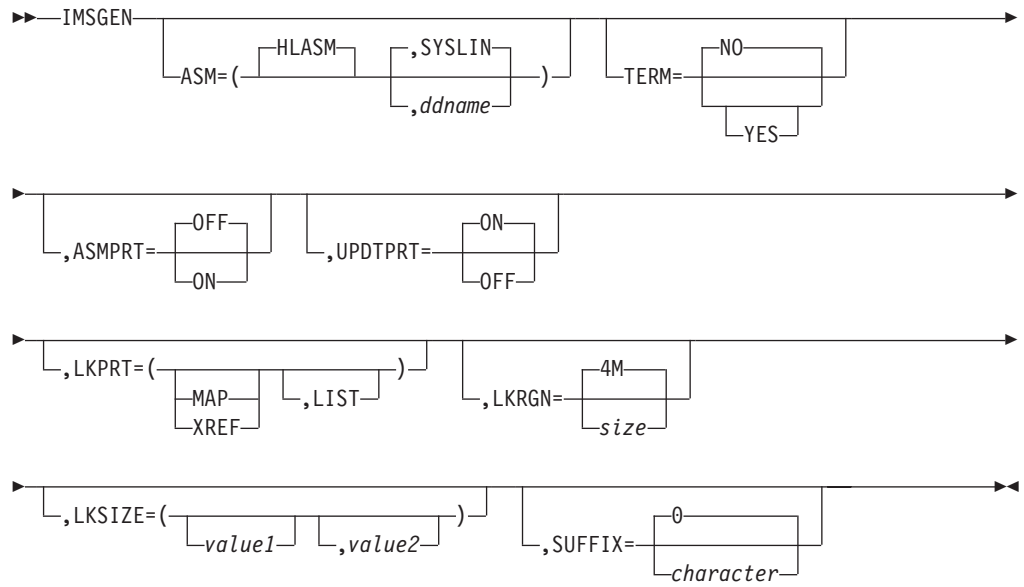
Specifies the one- to four-character name that becomes the instance name for the transport manager subsystem (TMS) and the subsystem name (SSN) for an RSR active or tracking subsystem. If TMINAME= is not specified, IMS uses a series of blanks as the default, which associates IMS with the TMS that used a default instance name.

---

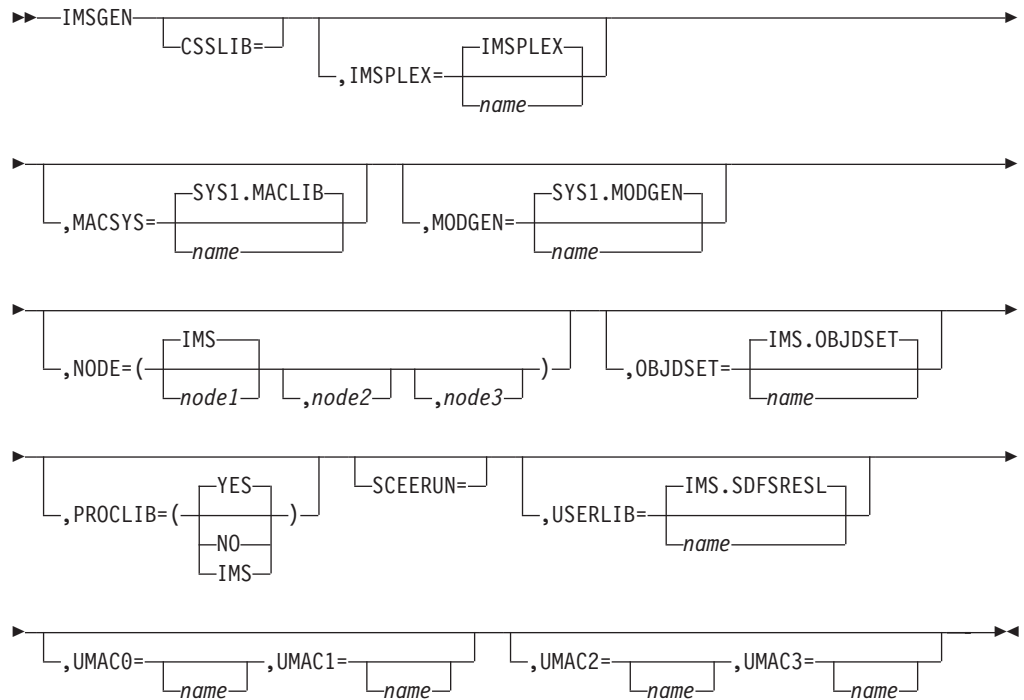
## IMSGEN Macro

IMSGEN specifies the assembler and binder data sets and options, and the system definition output options and features.

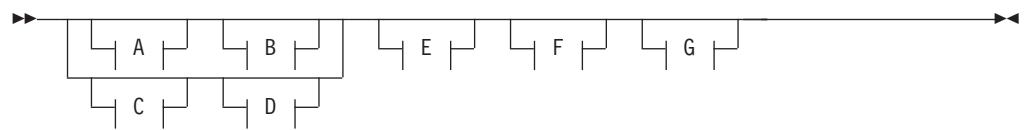
IMSGEN must be the last IMS system definition macro in the stage 1 input stream, and it must be followed by an assembler END statement.



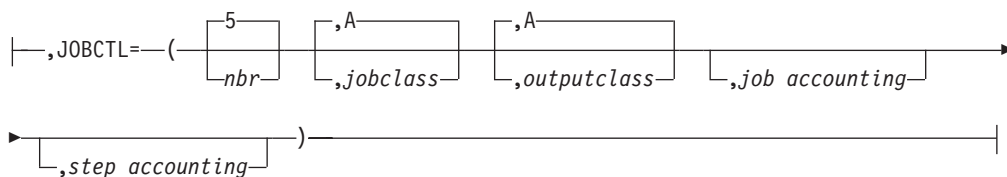
**IMS Data Set Options**



**Job Control Language Options**



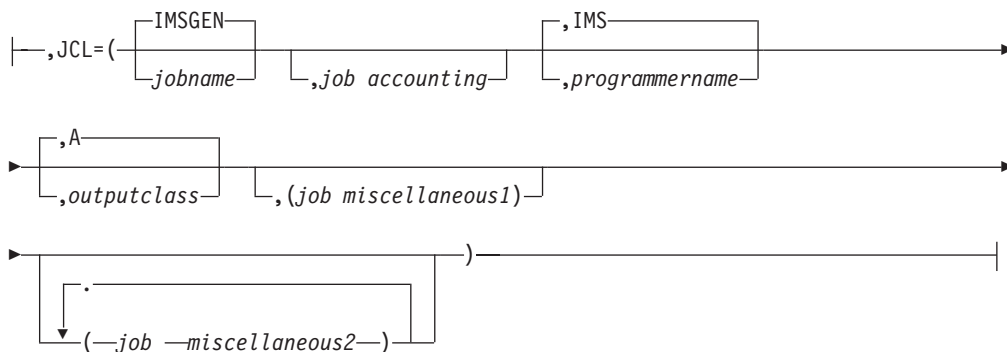
**A:**



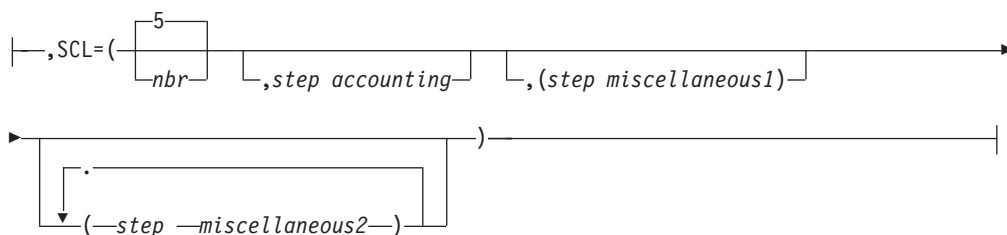
**B:**



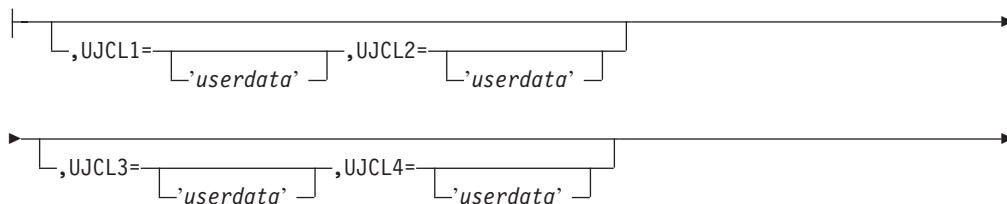
**C:**

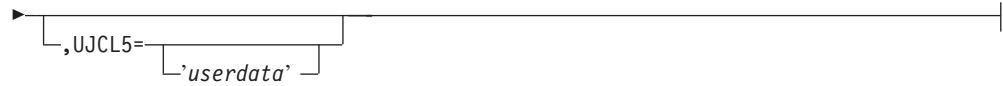


**D:**

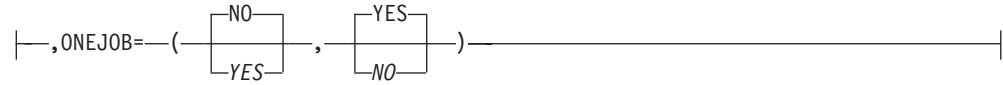


**E:**

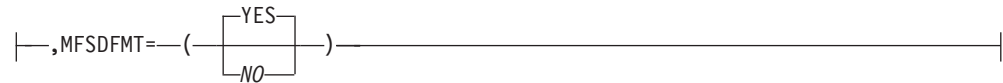




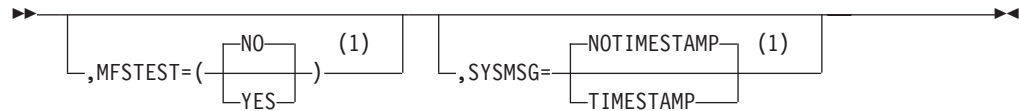
**F:**



**G:**



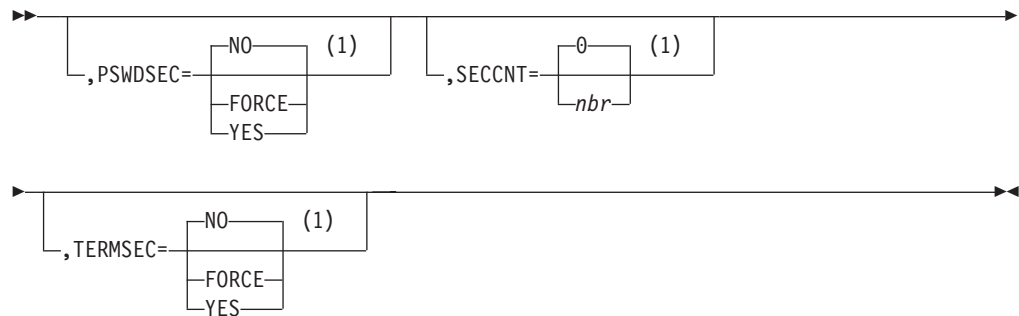
**General Communication Options**



**Notes:**

- 1 If you specify the COMM macro, the general communication option parameters are overridden.

**Security Options**



**Notes:**

- 1 If you specify the COMM or SECURITY macro, the security option parameters are overridden.

**Keyword Parameters**

To find which parameters apply to your IMS configuration, refer to Table 15 on page 33.

## Assembler and Binder Options

### ASM=

The first parameter specifies the assembler JCL to be produced for the stage 2 assembly steps. HLASM, the high level assembler, is the only valid first parameter.

For an LGEN assembly, the high level assembler is used in the stage one assembly cycles.

The second parameter can be used to specify the ddname to be used for the assembler object output. If the ddname is not specified, the ddname is SYSLIN.

Avoid the use of ddnames other than SYSLIN, SYSGO, and SYSPUNCH if you are using the z/OS System Modification Program (SMP) to apply maintenance to your IMS libraries. SMP does not recognize other ddnames during JCLIN processing.

The option of changing the object output ddname is intended to accommodate those users who have installed an assembler and altered its ddname defaults. The stage two assembler execution statement options are PARM='OBJECT,NODECK,NODBCS'. The object file output ddname is the one that must be coded in place of the default value of SYSLIN. Also, you must use the default assembler option ALIGN for assemblies of IMS modules.

**Recommendation:** Use NOUSING and FLAG(NOPUSH) options to avoid assembler errors. Because the assembler code that is created might contain lowercase characters, you should not specify the assembler option COMPAT(CASE).

### ASMPRT=

Specifies the assembler print options for those assembler job steps produced by IMS system definition.

ON specifies that assembly listings are to be generated.

OFF specifies that assembly listings for inline assemblies are not to be generated. The default is OFF.

### LKPRT=

Specifies binder print options for those binder job steps produced by IMS system definition. These print options include:

#### Value Print Option

**MAP** Module map

**XREF** Cross-reference table (XREF includes the MAP option)

**LIST** List of control statements in statement image format

If this parameter is omitted, only linkage editor error messages, if any, are printed.

### LKRGN=

Specifies a region size value to be placed on the generated EXEC statements for execution of the binder. This parameter can be specified in kilobytes or megabytes.

In kilobytes, the value can be 1 through 7 decimal numbers (1 through 2096128).

In megabytes, the value can be 1 through 4 decimal numbers (1 through 2047).



**LKSIZE=**

Specifies the values to be placed in the SIZE parameter of the EXEC statements for the binder job steps.

**value1**

Specifies the maximum number of bytes of main storage and available virtual storage. This value can be specified in the form *n* (where *n* represents the actual number of bytes of virtual storage, not to exceed 16384000) or *nK* (where *n* represents the number of 1 KB blocks of virtual storage, not to exceed 16000K).

**value2**

Specifies the number of bytes of storage to be allocated for the load module buffer. It is expressed in the form *n* (where *n* cannot exceed 65520) or *nK* (which cannot exceed 63K).

Default values are not supplied in LKSIZE for the binder SIZE parameter. If *value1* and *value2* are omitted, the Binder SIZE defaults are used.

**Related Reading:** For a more detailed description of the SIZE option, see *DFSMS/MVS Program Management*.

**SUFFIX=**

Specifies the alphanumeric suffix character appended to the generated composite control block, nucleus, MFS device characteristics table, and security directory module names when they are put into the IMS.SDFSRESL or IMS.MODBLKS data sets. The IMS online nucleus name always starts with DFSVNUC. The suffix character supplies the eighth character of the nucleus names. If the suffix character equals 0, the online nucleus name is DFSVNUC0. This concept allows the system user to generate multiple IMS systems for use in one environment where the characteristics of each system vary. The default is 0.

The modules that receive the suffix are<sup>2</sup>:

DBFSCD0x, DBFSCD1x	DFSISDBx
DFSBLK0x	DFSISDCx
DFSCLCxx	DFSOPLOx
DFSCCLIDx	DFSPDIRx
DFSCLL0x	DFSRCTEx
DFSCLL1x	DFSRSR0x
DFSCLRxx	DFSSCD0x, DFSSCD1x
DFSCLSxx	DFSSMB0x
DFSCLVxx	DFSSYS0x
DFSDDIRx	DFSUDT0x
DFSFEDBx	DFSVNUCx
DFSFRB0x, DFSFRB1x	

The suffix is also appended to the intermediate control block assemblies for SMP/E purposes.

**TERM=**

Specifies the assembler TERM/NOTERM option for stage 2 assembler job steps produced by the IMS system definition.

2. If no IMSGEN is coded with the default SUFFIX of 0, subsequent functions might fail, from the lack of the default, DFSUDT0 module

YES specifies that the TERM option is in the PARM= field for each of the stage 2 assembler job steps. YES also specifies that the //SYSTEM DD statement is to be built in each of the stage 2 assembler job steps.

NO specifies that the NOTERM option is included in the PARM= for each of the stage 2 assembler job steps. The default is NO.

**UPDTPRT=**

Specifies whether the IEBUPDTE SYSPRINT is to be produced by IMS system definition stage 2.

ON specifies that the IEBUPDTE SYSPRINT is to be generated. The default is ON.

OFF specifies that the IEBUPDTE SYSPRINT is to be suppressed. This includes the suppression of any errors or warnings issued by the IEBUPDTE utility.

## IMS Data Set Options

**CSSLIB=**

Specifies the name of the OS/390 (R2V10 or above) callable services library data set that will be part of the concatenation of the STEPLIB DD statement of the DFSJMP and DFSJBP procedures. The specified data set name cannot be longer than 44 alphanumeric characters. If the keyword is not specified, a default value of SYS1.CSSLIB will be used as the data set name.

No system definition is needed unless you will be using the DFSJMP or DFSJBP proc lib members created by system definition.

**IMSPLEX=**

Specifies the data set qualifier (or qualifiers) for the OLCSTAT data set. This name should be the same for all IMS systems running in the same IMSplex because they will all share the same OLCSTAT data set.

The name specified on the IMSPLEX option can be from one to 35 characters in length (including embedded periods). The first character of each level of qualification must be a letter, @, \$, #, or a number (zero through 9). The IMSPLEX name must conform to OS/390 data set naming conventions. The default name is IMSPLEX.

**MACLIB=**

This keyword is obsolete.

**MACSYS=**

Specifies the name for the MACLIB data set to be used in the stage 2 assemblies. The specified data set name cannot be longer than 44 alphanumeric characters.

**MODGEN=**

Specifies the name for the MODGEN data set to be used in the stage 2 assemblies. The specified data set name cannot be longer than 44 alphanumeric characters.

**NODE=**

Specifies the node assigned to IMS data set names.

**node1**

Specifies the node to be assigned to all IMS data set names, as related to the use and definition of the data set names, by IMS system definition. The specified node can contain from 1 to 26 characters, including imbedded periods. The first character of each level of qualification must be a letter, @,

\$, or #. The remaining seven characters can be any of the preceding characters and the numerals zero through nine. The node name must conform to z/OS data set naming conventions. The default node generated is IMS.

### node2

Specifies the node to be assigned to the IMS data set names listed in Figure 12. This node overrides the *node1* assignment for these specific data sets. The character set restrictions are as defined for *node1*.

### node3

Specifies the node to be assigned to the IMS data set names listed in Figure 12. This node overrides the *node1* assignments for these specific data sets. The character set restrictions are as defined for *node1*.

node1	node1	node2	node3
MSDBCP1	IMSLOG2	PROCLIB	ADFSMAC
MSDBCP2	RDS	SDFSRESL	ADFSLOAD
MSDSCP3	RDS2	SDFSMAC	ADFSRC
MSDBCP4	PGMLIB	MATRIX	LGENIN
MSDBDUMP	PSBLIB	MATRIXA	LGENOUT
MSDBINIT	DBDLIB	MATRIXB	
QBLKS	ACBLIB	JOBS	
QBLKSL	ACBLIBA	OPTIONS	
SHMSG	ACBLIBB	MODBLKS	
SHMSG	REFERAL	MODBLKSA	
LGMSG	FORMAT	MODBLKSB	
LGMSG	FORMATA		
MODSTAT	FORMATB		
MODSTAT2	TFORMAT		
IMSMON	SYSO <sup>nnn</sup>		
IMSLOG			

Figure 12. Node Assignments for IMS Data Set Names

### OBJDSET=

Specifies the name of a cataloged partitioned data set into which assembler object modules are assembled during stage 2 of IMS system definition. The specified data set name must not be longer than 44 alphanumeric characters. If this parameter is not supplied, these modules are placed in IMS.OBJDSET.

### PROCLIB=

Specifies whether system procedures are to be generated. If PROCLIB=NO, no system procedures are generated. If PROCLIB=NO, a new definition does not reflect the proper defaults because the newly generated procedures are not stored in PROCLIB. Therefore, care should be taken when specifying this option.

If PROCLIB=IMS and a CTLBLKS, NUCLEUS, or ON-LINE definition is specified in the IMSCTRL macro statement, only the IMS procedure is generated. If PROCLIB=IMS is specified and a BATCH or ALL definition is specified, the PROCLIB=IMS specification is ignored and the default (YES) is used.

The second parameter (n) is no longer supported by the system definition process. It was used to define the OSAM buffer pool size in 1K increments for a

DL/I batch region. If coded for compatibility, it is accepted and entered as the BUF= parameter on the DBBATCH, DLIBATCH, IMSCOBGO, and IMSPLIGO cataloged procedures.

If the second parameter is not coded, a default value of 7K bytes is used. BUF= is generated in the cataloged procedures and entered as the BUF= parameter in the DBBATCH, DLIBATCH, IMSCOBGO, IMSPLIGO catalog procedures.

For information on specifying database buffer pool size batch region for a DL/I batch region, see "Specifying IMS Buffer Pools" on page 430.

**SCEERUN=**

Specifies the name of the System C Runtime Library data set that will be part of the concatenation of the STPLIB DD statement of the DFSJMP and DFSJBP procedures. The specified data set name cannot be longer than 44 alphanumeric characters. If the keyword is not specified, a default value of CEE.SCEERUN will be used as the data set name.

**USERLIB=**

Specifies the name of the library in which user-written routines, such as message edit routines, are to be included in the generated IMS nucleus. If this operand is omitted, the library containing the routines is assumed to be node(2).SDFSRESL data set. The name of the data set, which must be cataloged, must not be longer than 44 characters.

**UMAC0=**

Specifies the name of a data set to be put at the top of the SYSLIB concatenation of the stage 2 assembly job steps. The specified data set name cannot be longer than 44 alphanumeric characters.

**UMAC1=, UMAC2=, and UMAC3=**

Specifies the name of a data set to be concatenated to the SYSLIB DD statement of the stage 2 assembly job steps. The specified data set name cannot be longer than 44 alphanumeric characters.

Typically MACSYS is used for either SYS1.MACLIB or SYS1.AMACLIB, and MODGEN is used for either SYS1.MODGEN or SYS1.AMODGEN. If the SYS1.AMACLIB and SYS1.AMODGEN data sets are specified, the UMAC1= keyword must specify SYS1.ATSOMAC. This insures that all macros that might be referenced are at the same level.

If no value is provided, the corresponding statement is dropped from the SYSLIB DD.

## Job Control Language Statements

Use the first two parameters (JOBCTL and PRTY) or the last two (JCL and SCL).

**JOBCTL=**

The following parameters are valid:

**nbr**

Specifies the number of steps per JOB statement to be produced by system definition stage 1 for execution of stage 2. The maximum allowable value is 255; the default is 5. Regardless of the value specified, a JOB statement is produced for the beginning of the bind steps and for the nucleus bind step (if applicable).

**jobclass**

Specifies the job class to be generated on the stage 2 JOB statement. The default is A.

**outputclass**

Specifies the output class to be generated for the stage 2 JCL. The default is A.

**job accounting and/or step accounting**

Specifies job and/or step accounting data to be placed in the stage 2 JCL. The length of each set of (job and step) accounting data cannot exceed 50 bytes. If job accounting data is specified, a programmer name of IMS is provided.

When the accounting information fields contain special characters (except hyphens), special considerations apply to the manner in which this field is specified. See "Sample IMSGEN Macro Statement" on page 110.

**JPRTY=**

Specifies the priority placed on the JOB statements for IMS system definition stage 2 jobs. The default is priority 0.

**JCL=**

The following parameters are valid:

**jobname**

Specifies a maximum of six alphanumeric characters to be used as the first portion of the generated job names. The last two characters of the job names are internally generated, sequentially incremented, hexadecimal numeric values representing the relative position of each job in the stage 2 stream. The default is IMSGEN.

The character set that can be used is described on the node keyword.

**job accounting**

Specifies job accounting data to be placed in the stage 2 JCL. The length of the accounting data cannot exceed 50 bytes.

When the accounting information fields contain special characters (except hyphens), special considerations apply to the manner in which this field is specified. See "Sample IMSGEN Macro Statement" on page 110.

**programmername**

Specifies the programmer name to be placed in the stage 2 JCL. The default is IMS.

**outputclass**

Specifies the output class to be generated for the stage 2 JCL. The default is A.

**job miscellaneous 1 & 2**

Specifies any additional parameters the user can desire to have placed in the stage 2 JOB statements. Length of this parameter cannot exceed 60 bytes. If you code this parameter, do not code the MSGCLASS parameter.

**MFSDFMT=**

Specifies whether (YES) or not (NO) the default message format screens are built in stage 2 of IMS system definition. If MFSDFMT=NO, stage 2 does not have the default MFS screens generated in stage 2. The default is YES.

**ONEJOB=**

Subparameter1 specifies whether (YES) or not (NO) a single job is to be built for stage 2. The default is NO. If ONEJOB=NO, stage 2 has a multiple of job steps as specified in JOBCTL= subparameter 1.

Subparameter 2 specifies whether (YES) or not (NO) condition code checking for the bind steps is to be performed. If ONEJOB=(YES,YES) is specified,

condition code checking is done in the bind steps. These steps are bypassed if any previous step did not complete with a condition code of 0. The default for subparameter 2 is YES if subparameter 1 is YES.

When ONEJOB= is specified for a single JOB statement, JOBCTL= and SCL= subparameter 1 are ignored.

If condition code checking is being specified for SCL= and ONEJOB= subparameter 2 is YES, a JCL error occurs due to the specification of duplicate COND= parameters.

### **SCL=**

The following parameters are valid:

#### **nbr**

Specifies the number of steps per JOB statement to be produced by system definition stage 1 for execution of stage 2. The maximum allowable value is 255; the default is 5. Regardless of the value specified, a JOB statement is produced for the beginning of the bind steps and for the nucleus bind step (if applicable). This parameter is ignored in two cases:

- The first job that updates the IMS options data set or the IMS PROCLIB has one or two job steps for system definitions (one job step for each applicable update). IMS requires this for compatibility with the SMP/E GENERATE function.
- The last job is a multi-step job with two or four steps (for system definitions that create MFS libraries. This is caused by steps 2 and 4 (if present) being dependent on the respective preceding step.

#### **step accounting**

Specifies step accounting data to be placed in the stage 2 JCL. The length of the accounting data cannot exceed 50 bytes.

When the accounting information fields contain special characters (except hyphens), special considerations apply to the manner in which this field is specified. See "Sample IMSGEN Macro Statement" on page 110.

#### **step miscellaneous**

Specifies any additional parameters that you might want to place in the stage 2 EXEC statements. The length of this parameter cannot exceed 50 bytes. Do not code REGION= here, because internal IMS processing automatically generates this parameter. Do not code COND= if the MFSDfmt=NO keyword is not coded. Some of the default MFS job steps have COND= already coded.

### **UJCL1=,...,UJCL5=**

Specifies the contents of a JCL statement that is placed between the JOB statement and the first EXEC statement in the stage 2 job stream. The specified statement cannot be longer than 60 characters. No other syntax checking of the parameter contents is done.

The intent of these keywords is to provide comment or JES control statements in the stage 2 job stream. These parameters can be used, for example, to add ROUTE statements, JOBLIB DDs, or JOBCAT DDs.

If no value is provided, the corresponding statement is dropped from the SYSLIB DD.

## General Communication Options

All communication operands specified are ignored and a warning message is issued if the COMM macro is specified. See the section entitled “Relationships between the SECURITY, COMM, and IMSGEN Macros” on page 136.

### MFSTEST=

Specifies whether (YES) or not (NO) the Message Format Service test facility, MFSTEST, is to be included in the generated system. YES is invalid for systems not containing MFS terminals. The default is NO.

Be aware that the use of MFSTEST can degrade IMS performance due to MFSTEST use of the communication line buffer pool.

### SYSMSG=

Specifies whether (TIMESTAMP) or not (NOTIMESTAMP) the system message timestamp facility is to be included for the generated system. The default is NOTIMESTAMP.

When TIMESTAMP is specified, the time that a message was generated is inserted between the message number and the message text for each message in the table.

**Related Reading:** Refer to the DFS messages introduction in *IMS Version 9: Messages and Codes, Volume 2* for more information.

## Security Options

All security options are ignored or overridden if the COMM or the SECURITY macro is specified.

**Related Reading:** See “Relationships between the SECURITY, COMM, and IMSGEN Macros” on page 136 for more information about security options.

### PSWDSEC=

Specifies whether (YES or FORCE) or not (NO) password security is to be generated as the default for the /NRESTART command when issued to start the IMS system. The default is NO.

FORCE specifies that the IMS MTO cannot negate password security when issuing the /NRESTART command. If YES is specified, the MTO can override password security inclusion by specifically requesting no security on the /NRESTART command.

### SECCNT=

Specifies the maximum number of terminal and password security violations to be accepted per physical terminal and transaction command violations per transaction prior to master terminal notification of such violation. The default is 0, which nullifies notification to the master terminal. The number specified can range from 0 through 3.

If SECCNT is not 0, the master terminal is notified for every violation.

### TERMSEC=

Specifies whether (YES or FORCE) or not (NO) terminal security is to be generated as the default for the /NRESTART command when issued to start the IMS system. The default is NO.

FORCE specifies that the IMS MTO cannot negate terminal security when issuing the /NRESTART command. If YES is specified, the MTO can override terminal security inclusion by specifically requesting no security on the /NRESTART command.

## Sample IMSGEN Macro Statement

The following example of the IMSGEN macro statement indicates that:

- High Level Assembler is used.
- An assembly listing is desired.
- Binder print options are MAP and LIST.
- SIZE parameter specification is not required.
- Binder region size is 4 MB.
- Composite control block, nucleus, map module, nucleus and security directory blocks module suffix is 9.
- The cataloged partitioned data set into which assembler load modules are to be placed is IMSXXX.OBJ.
- The library containing user routines is IMSXXX.USER.
- System procedures are requested.
- All IMS data set names are to be prefixed by the node IMSXXX (that is, IMSXXX.SDFSRESL).

```

IMSGEN      ASMPRT=ON,LKPRT=(MAP,LIST),SUFFIX=9,          72
              OBJDSET=IMSXXX.OBJ,                        X
              USERLIB=IMSXXX.USER,                       X
              PROCLIB=YES,NODE=IMSXXX                     X
    
```

The examples of the IMSGEN macro statement shown in Figure 13 compare the JOB and EXEC statements produced using the JOBCTL and PRTY keywords to the statements produced using the JCL and SCL keywords.

```

IMSGEN      JOBCTL=(4,D,A,(P01,9987)),PRTY=8              72

//IMSGEN1 JOB (P01,9987),IMS,MSGLEVEL=1,MSGCLASS=A,
              CLASS=D,PRTY=8

IMSGEN      JCL=(OSCAR,(P01,9987),FELIX,,(MSGLEVEL=1,    X
              PRTY=3,TYPRUN=HOLD))

//OSCAR01 JOB (P01,9987),FELIX,MSGLEVEL=1,MSGCLASS=A,
              PRTY=3,TYPRUN=HOLD

IMSGEN      JCL=(GEORG,(P01,"9/12/99"),IMS,D,(PRTY=3)), X
              SCL=(3,("/83468"),(PARM=123))

//GEOG01 JOB (P01,"9/12/99"),
// IMS,
// PRTY=3,
// MSGCLASS=D
//STEP1 EXEC PGM=IEBUPDTE,PARM=NEW,
//          ACCT=("/83468"),
//          PARM=123
    
```

Figure 13. Sample IMSGEN Macro Statements

When the accounting information consists of more than one subparameter, you must enclose the information in parentheses, not apostrophes—that is: (5438,GROUP6).

If any of the subparameters contains special characters (except hyphens), enclose that subparameter in double apostrophes, not quotation marks—that is:



(5438, '12/15/99:').

## LINE Macro

The LINE macro statement describes both switched and nonswitched communication lines. If the line described has only one terminal attached, only one TERMINAL macro instruction is included after the LINE macro instruction. Multiple TERMINAL macro instructions are included if the description is for a multidrop line. For nonswitched lines, multiple NAME macro instructions can be included after each TERMINAL macro instruction that follows a LINE macro instruction. Each LINE macro instruction must be followed by at least one TERMINAL macro instruction. Only one LINE statement per line group is allowed when UNITYPE=(3270,LOCAL) is specified on the LINEGRP macro statement.

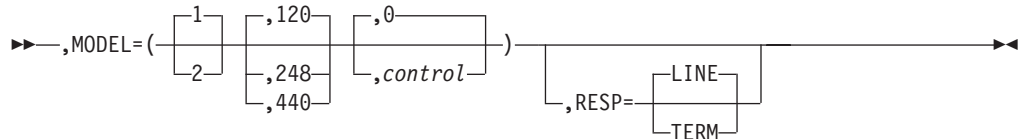
### All Lines



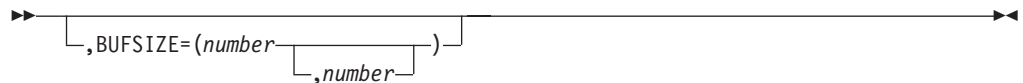
### All Lines Except Spool and 3270 Local



### 2740 Lines Only



### 3270 (Local), S/3, S/7, Disk Tape, and Spool Lines Only



### 3275 Switched Lines Only



## Keyword Parameters

To find which parameters apply to your IMS configuration, refer to Table 15 on page 33.

**ADDR=**

Specifies the address of the communication line as defined in the Transmission Control Unit. The address value is three or four hexadecimal digits between 0000 and FFFF. All line addresses specified in the system definition must be unique values. This operand is used only to generate IMS DD statements for cataloged procedures. It is not allowed if the terminal is a 3270 local. This operand is not required for SPOOL line groups.

This operand is ignored for UNITYPE=PRINTER when a JES spool data set (SYSOUT=X) is to be used rather than a local printer. When XRF is used, a second optional device address can be specified with the ADDR keyword. This operand defines the second address for a BTAM-supported IMS master or secondary master terminal. And, when specified, only the master or secondary master terminals can be in the line group.

**BACKUP=**

Specifies (for XRF only) the control of the automatic restart after takeover. Use only when HSB=YES is specified on the IMSCTRL macro.

X is a numeric integer from 1 (lowest) to 7 (highest), inclusive, that specifies the priorities for reestablishing the session. If the entire keyword or the operand is omitted, the default is 4. Specifying BACKUP=NO suppresses the automatic restart of the devices after a takeover.

**BUFSIZE=**

This operand is only valid for 3270 (LOCAL), DISK, TAPE, System/3, System/7, and SPOOL lines.

For 3270 local lines, this operand is used to specify the maximum size of an input message on this line. The operand is required for 3270 local lines to which 3277 terminals are attached and is not allowed for lines to which 3284 or 3286 printers are attached. The minimum specification is 300 bytes; the maximum specification is 32000 bytes.

For DISK, TAPE, or SPOOL lines, this operand is used to specify, in bytes, the maximum output buffer size to be used for the line. The operand is required. The minimum specification is 16; the maximum specification is 32760.

For System/3 and System/7 lines, this operand is used to specify the input and output buffer sizes to be used for this line. The first parameter is the input buffer size; the second parameter is the output buffer size. The operand is required. The minimum specification for either parameter is 50; maximum specification is 65000.

For System/3 and System/7 BSC lines, the input buffer size specified is doubled by IMS to allow handling of reverse interrupts.

For System/7 lines, the output buffer size specified is doubled by IMS to allow for conversion to pseudo-binary PTTC/EBCD unless the user specifies FEAT=PTTC/EBCD in the STATION macro. The input buffer size, however, is not automatically doubled by IMS. It is the responsibility of the user to ensure that the buffer size specified contains the data in the format specified by the STATION macro.

**CONFIG=**

Specifies the name of a CONFIG macro, previously defined in either this line group or in a previously defined line group, that describes the 3275 configuration to be used:

- When VERIFY=NO is specified and either no IDLIST macro is provided or one or more of the list entries does not name a configuration to use.

- When VERIFY=YES is specified and one or more of the list entries does not name a configuration to use.

If CONFIG= is not specified, the IMS default configuration is used unless IDS= is specified.

**IDS=**

Specifies the name of the IDLIST macro, previously defined either in this line group or a previously defined line group, that is to be used for this line. When IDS= is used, no subsequent IDLIST macro need be specified for this line. The CONFIG specification from the named IDLIST macro's associated LINE macro is used unless this line also specifies CONFIG=.

**MODEL=**

Specifies (for 2740 line groups only) the model. The first parameter specifies the model number of the terminals attached to this line. If MODEL=1 is specified, all other parameters of the operand must be omitted. MODEL=2 is invalid for switched 2740 line groups. The default is 1.

The second parameter specifies the maximum input buffer size for all 2740 Model 2 terminals on this line. The default value is 120.

The third parameter specifies the header control feature for 2740 Model 2 terminals on this line, and the number of characters reserved for header control. The default value is 0. The value for header control must not exceed 28, and it must be a multiple of 4.

**RESP=**

Specifies that 2740 station control and 2740 Model 2 terminals are to operate in either line (LINE) or terminal (TERM) response mode. All terminals that are eligible to enter Fast Path transactions must operate in terminal response mode (TERM).

**VERIFY=**

Specifies whether (YES) or not (NO) terminal hardware ID verification should be performed when IMS receives a dialed-in call from a switched 3275. The default is NO. If VERIFY=NO, IMS checks for the presence of a list of terminal IDs (see "IDLIST Macro" on page 84). If no list is found, IMS sets up the configuration specified in the LINE macro (CONFIG=); if CONFIG= is not specified, IMS sets up the default configuration (see "CONFIG Macro" on page 78). If IMS finds an ID list, it searches the list for the calling terminal. If no match is found, IMS sets up the configuration specified in the LINE macro or the IMS default. If a match is found, IMS sets up the configuration accordingly.

If VERIFY=YES is specified, IMS searches the list of terminal IDs (established by the IDLIST macro) to determine if the calling terminal is authorized to access IMS using this line. If IMS finds a match, it sets up the configuration associated with this terminal (see "CONFIG Macro" on page 78). If IMS does not find a match, it disconnects the terminal, sends the DFS2106 UNKNOWN TERMINAL TRIED TO CONNECT ON LINE X PTERM Y message to the master terminal, and creates a X'69' log record containing the terminal ID and the number of the accessed line.

After the configuration is established, IMS sends a DFS073 /IAM COMMAND REQUIRED message to the calling terminal. The /IAM command is required to establish a logical connection to IMS.

**Example:** An example of the LINE and LINEGRP macros used in combination to define a 2740 Model 2 with maximum input buffer of 248 bytes, line address of 024, and no header control or edit routine is:

```
LINEGRP DDNAME=D27402,UNITYTYPE=2740
LINE ADDR=024,MODEL=(2,248)
```

**Use of SYSIN Local Card Reader:** When the DD statement for a line (specified in the LINEGRP macro statement as UNITYTYPE=READER) does not allocate a card reader device, the following conditions apply:

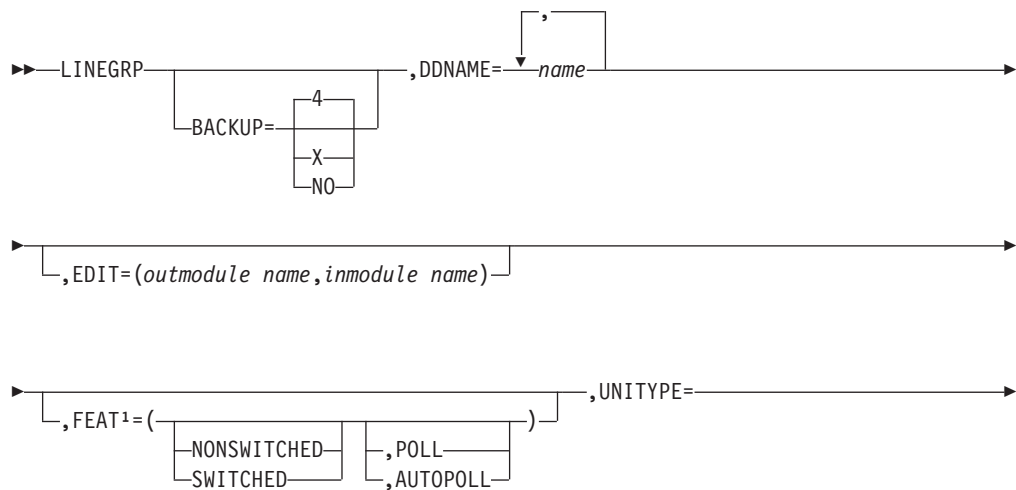
- If IMS is running as a batch program, the DD statement referring to the line must contain DCB=BLKSIZE=80.
- If IMS is running as a system task, the SYSIN test stream must be placed in a sequential data set with the DCB attributes RECFM=F and BLKSIZE=80. The SYSIN test stream must be pointed to by a DD statement for a reader line within the IMS procedure.

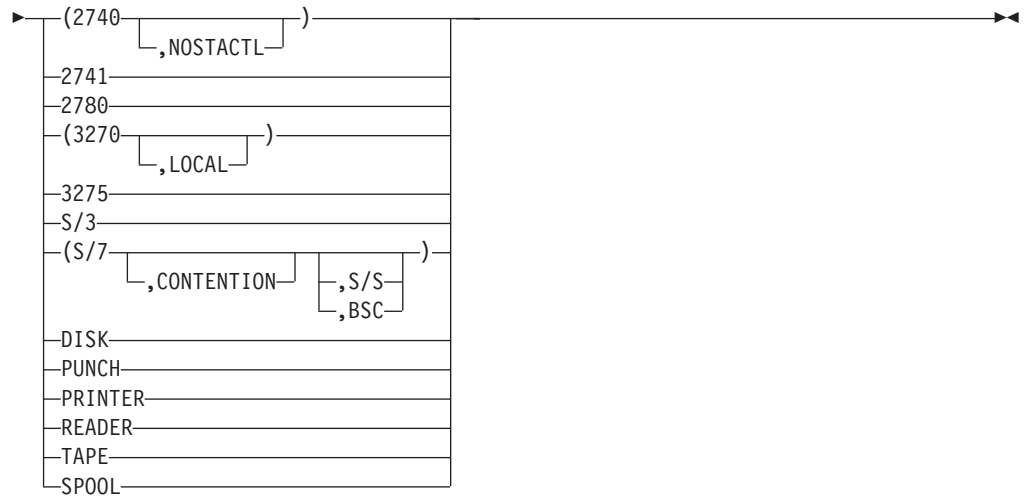
Although VTAM requests are priority-ordered by IMS, the active requests can be completed in any order because of internal VTAM conflicts and pacing.

## LINEGRP Macro

The LINEGRP macro statement defines the beginning of a set of macro instructions that describe the user's telecommunications system. This macro statement begins a description of one or more lines of the same type over which the same type of terminal communicates.

### All Line Groups

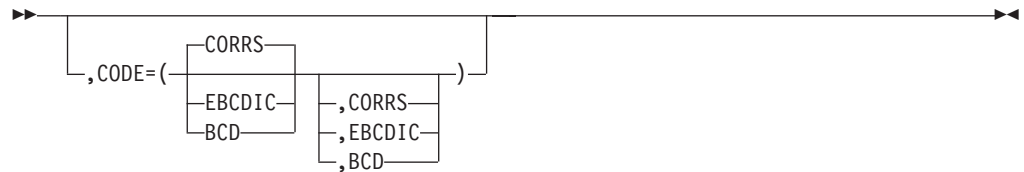




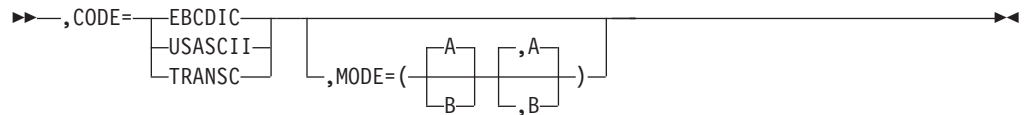
**2741 Nonswitched Line Groups Only**



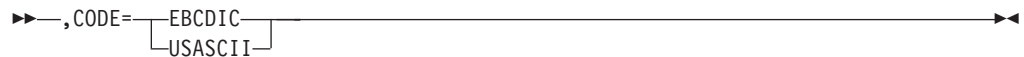
**2741 Switched Line Groups Only**



**2780 Line Groups Only**



**3270 Remote and 3275 Switched Line Groups Only**



**Keyword Parameters**

To find which parameters apply to your IMS configuration, refer to Table 15 on page 33.

**BACKUP=**

Specifies (for XRF only) the control of automatic restart after takeover. Use only when HSB=YES is specified on the IMSCTRL macro.

X is a numeric integer from 1 to 7, inclusive, which specifies the priority assigned to reestablishing the session. When either the keyword or the operand is omitted, the default is 4. Specifying BACKUP=NO suppresses the automatic restart of the devices after a takeover.

**CODE=**

Specifies translate options for 2780 line groups, 2741 line groups, and 3270 remote line groups only.

For 2780 line groups, specifies translate table EBCDIC/EBCDIC (EBCDIC), RASA/SASA (USASCII), or RC80/SD80 (TRANSC). The operand is required.

For nonswitched 2741 line groups, specifies translate table SD40/RC40 (EBCDIC), SB40/RB40 (BCD), or SD41/RC41 (CORRS). The default is CORRS. The operand is required.

For switched 2741 line groups, specifies one or a combination of translate tables. Actual translate table determination is performed when the terminal calls IMS. The combination of EBCDIC and BCD is invalid. The default is CORRS (RC41/SD41).

**DDNAME=**

Specifies a one- to eight-character name that associates the generated DCB for this line group, within the IMS nucleus control block, with the generated JCL DD statements identifying the lines within the line group. If the line group being defined is a SPOOL line group, a list of up to 20 names can be specified: 255 SPOOL data sets, but no more than 20 names per line group can be specified. Each SPOOL name specification is associated, in the order in which it was specified, with the sequentially generated SPOOL data set names produced for the SPOOL print procedures. These procedures are generated and named sequentially for each SPOOL line group. Refer to "LINEGRP Macro" on page 114, for an example of SPOOL data definition name association.

The names must begin with an alphabetic character. The operand is required. The following names cannot be used as LINEGRP ddnames:

DFSRESLB	IMSLOG	IMSTFMT	MSDBINIT
DUMP	IMSLOGR	IMSUDUMP	PRINTDD
IEFRDER	IMSLOGR2	LGMSG	PROCLIB
IEFRDER2	IMSLOG2	MATRIX	QBLKS
IMSACB	IMSMON	MSDBCP1	SHMSG
IMSDBL	IMSRDS	MSDBCP2	
IMSDILIB	IMSSPA	MSDBDUMP	

**Recommendation:** Allocate at least two data sets.

**EDIT=**

Specifies a one- to eight-character, user-supplied physical terminal output edit routine and a one- to eight-character, user-supplied physical terminal input edit routine for the terminal type in this line group.

**Related Reading:** For more information, refer to *IMS Version 9: Customization Guide*.

This routine cannot be the same as the one that is used on a TRANSACT EDIT= parameter.

**FEAT=**

Determines whether this group of telecommunication lines is leased

(NONSWITCHED) or not leased (SWITCHED). The default is NONSWITCHED, except for devices that are not valid nonswitched devices. SWITCHED is forced for 3275<sup>3</sup>

IMS supports the IBM Communicating Magnetic Card/ELECTRIC Typewriter (CMC/ST) as a switched 2741. Therefore, when a CMC/ST is used, it must be defined to IMS as a switched 2741. CMC/ST features that are not compatible with 2741 features, and that result in different operations, are not supported.

Polling features are ignored for READERS, PRINTERS, PUNCHes, TAPES, DISKS, SPOOLs, System/3s, and 3270s. The AUTOPOLL polling feature is invalid for 3270s (LOCAL), 2741s, and 2740s (NOSTACTL). No polling list is generated for 3270s (LOCAL), 2740s (NOSTACTL), 2741s, READERS, PRINTERS, PUNCHes, TAPES, DISKS, or SPOOLs. A BTAM AUTOWLST is generated for 3270s and multipoint 2780s. No polling list is generated for point-to-point 2780s unless specifically requested by specifying AUTOPOLL.

Direct SYSOUT specifications (PRINTER, PUNCH, TAPE, or DISK) do not require that specific device type be assigned at IMS execution time. All specifications except PRINTER affect only default BUFSIZE or generated JCL. Specifying PRINTER results in execution time being translated to the 48-character set, lowercase is translated to uppercase, and all other codes translated to periods(.). If a specification other than PRINTER is used, no translation occurs. If a line not generated as a PRINTER is allocated to a printer having the universal character set feature (UCS), and fold mode operation is used, unprintable characters print as extraneous alphanumerics.

#### **MODE=**

The first parameter specifies that communication is to be through the 270x Data Adapter Unit's Dual Communication Interface.

The second parameter specifies the use of the transmission code designated by Code (A or B) for the 270x Data Adapter Unit Dual Code Feature. This feature must be present on the 270x. A is the default.

#### **UNITYPE=**

Specifies the terminal device type contained in this line group. If UNITYPE=(3270,LOCAL) is specified, only one LINE statement can be in the line group.

CONTENTION (as opposed to a polled line) is a valid specification for System/7 only. Either S/S (start/stop, the default) or BSC (binary synchronous) can be specified.

When SWITCHED is specified for FEAT=, only 2740 with station control, 2741, and 3275 are valid for the UNITYPE= parameter.<sup>3</sup>

When FEAT=NONSWITCHED, valid UNITYPE parameters are 2740 with or without station control, 2741, 2780, 3270 local and remote, S/3, S/7, DISK, PUNCH, PRINTER, READER, TAPE, and SPOOL.

The following UNITYPE parameters do not cause BTAM to be included in the IMS system: DISK, PUNCH, PRINTER, READER, TAPE, or SPOOL.

When a JES SPOOL DATASET (SYSOUT) is used, UNITYPE = PRINTER must be specified.

---

3. A 3275 on a multidropped line is defined on the LINEGRP macro statement as UNITYPE=3270 and on the TERMINAL macro statement as UNIT=3275.

**Example:** Following is an example of the LINEGRP macro for a 2780 line group with no output edit routine and EBCDIC translation:

```
LINEGRP DDNAME=DD2780,UNITYPE=2780,CODE=EBCDIC
```

An example of the LINEGRP macro and the resulting ddname/dsname and print procedure associations for a SPOOL line group is:

```
LINEGRP DDNAME=(NAME1,NAME2),UNITYPE=SPOOL
```

```
LINEGRP DDNAME=(NAMEA,NAMEB),UNITYPE=SPOOL
```

The SPOOL print procedure for line group 1 is named IMSWT000. The data set names produced for this print procedure are IMS.SYSO1 and IMS.SYSO2.

The SPOOL print procedure for line group 2 is named IMSWT001. The data set names produced for this print procedure are IMS.SYSO3 and IMS.SYSO4.

The corresponding JCL statements generated for the IMS online execution procedure are as follows:

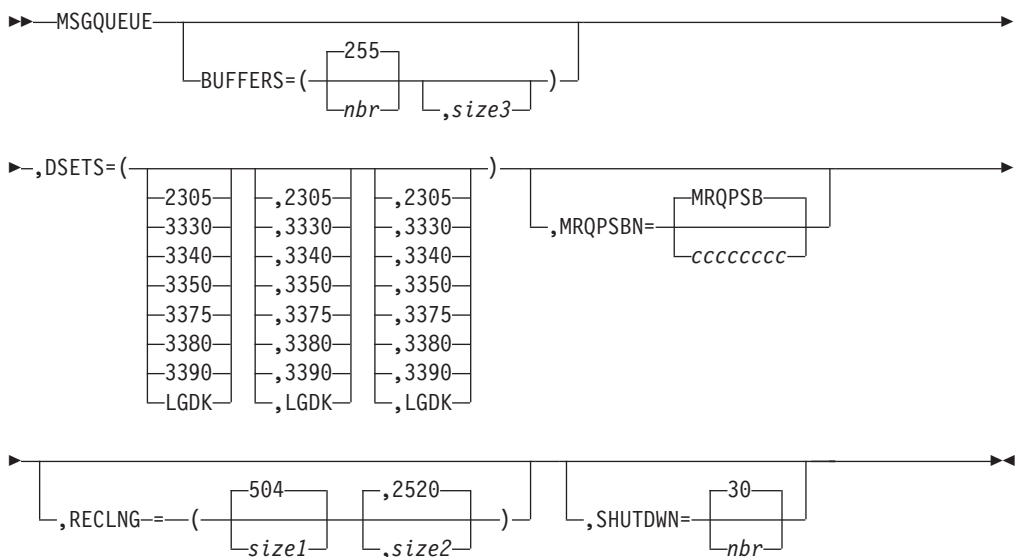
```
//NAME1 DD DISP=SHR,DSNAME=IMS.SYSO1
//NAME2 DD DISP=SHR,DSNAME=IMS.SYSO2
//NAMEA DD DISP=SHR,DSNAME=IMS.SYSO3
//NAMEB DD DISP=SHR,DSNAME=IMS.SYSO4
```

If UNITYPE=SPOOL on the LINEGRP macro statement, ddnames used in the preceding JCL must agree with the actual ddnames specified on this statement.

## MSGQUEUE Macro

You must include the MSGQUEUE macro statement when the type of definition specified in the IMSCTRL macro statement is ALL, ON-LINE, CTLBLKS, or NUCLEUS. This statement defines the characteristics of the three message queue data sets (IMS.QBLKS, IMS.SHMSG, and IMS.LGMSG).

The information you specify in this macro is also used in a shared queues environment.





## Keyword Parameters

To find which parameters apply to your IMS configuration, refer to Table 15 on page 33.

### **BUFFERS=**

Specifies the number of buffers to allocate for message queue management, and the block size to be assigned for all three message queue data sets. If *nbr* is not specified, a default of 255 is used. The minimum *nbr* specification is 3; the maximum is 9999. At execution time, the minimum value for buffers in a shared queues environment is increased to 200. If *size3* is not specified, it is calculated by the following formula:

$$SIZE3 = ((SIZE1+SIZE2-1)/SIZE1) \times ((SIZE1+55)/56) \times 56$$

For the division operations in this formula, only the integer portion of the result is used; the fractional portion of the result is truncated.

This calculation can leave null space at the end of the buffer block for short and long message records.

*size1* and *size2* should be an even multiple of the QBLKS LRECL of 56.

If the RECLNG defaults of *size1* (504) and *size2* (2520) are used, then *size3* of BUFFERS equals 2520. The maximum specification for *size3* is 30632, or the track length of the device on which the data set resides, whichever is smaller. All sizes specified are rounded up to multiples of 4.

RECLNG can be changed prior to a cold start of IMS, but if a smaller value is specified prior to a BLDQ restart, the restart might abend.

You should not allocate less space for a restart of IMS than was allocated for the prior execution. If a smaller logical record length has been allocated, or if less space for the Message Queue data sets has been allocated for restart with BUILDQ, the restart process may terminate abnormally.

### **DSETS=**

Specifies the device types on which the three message queue data sets (IMS.QBLKS, IMS.SHMSG, and IMS.LGMSG, respectively) are to reside. If all three data sets are to reside on the same device type, you can specify just the first parameter.

You can specify the following device types: 2305, 3330, 3340, 3350, 3375, 3380, 3390, or LGDK.

When using a 3350, specify the drive format used—3330 or 3350.

LGDK is a generic definition for disk drives that have a track size equal to or greater than 32767 bytes. The 3375, 3380, 3390, and future devices can also be defined as LGDK.

### **MRQPSBN=**

Specifies the one- to eight-character alphanumeric name of the Queue Control Facility (QCF) Program specification Block. Specifying this parameter permits IMS to use the support provided by the Queue Control Facility (Program 5699-E97) licensed program. The default name is MRQPSB. Regardless of whether you use the default name, you must include a PSB for the message region in the stage 1 system definition. If you do not use the default name, you

must specify a name for the MRQPSBN block with the MRQPSBN keyword. If an error is made in specifying the MRQPSBN PSB name, a warning message is issued and the default name is used.

**Note:** The MRQPSBN block is for the exclusive use of MRQ and QCF programs. It cannot be used by other user application programs running as MPPs, BMPs, or IFPs. If an attempt is made to use the PSB block by some other user application program, IMS will return a status code of MR and AIBRETRN code of 000000F0 on the call.

**Related Reading:** See the *IMS Version 9: Messages and Codes, Volume 1* for more information on this STATUS and AIBREASN code.

### RECLNG=

Specifies the logical record lengths for the short and long message queue data sets, respectively.

The minimum value for *size1* is 392. This size includes user data length of 64 bytes plus minimum prefix size of 328. The value specified for *size1* must be large enough to accommodate the ISC/LU6.1 (type 84) prefix and the Conversation Extension (type 8D) prefix. If Shared Queues is used, it must also include an additional X'28' bytes. If *size1* is not specified, a default of 504 is assigned.

The minimum value for *size2* is 1176. *size2* must be equal to or greater than *size1*. If *size2* is not specified, a default of 2520 is assigned. The value of *size2* cannot exceed the track length of the device on which the data set resides.

The *size2* value must be divisible by 4; otherwise the MSGQUEUE macro rounds it up until it is. *size1* and *size2* values should be even multiples of the QBLKS LRECL of 56.

If a message does not fit in a large queue record (*size2*), IMS either spans it to other queue records, or rejects the message with an error.

Some messages (such as MFS) from an input terminal can span message queue records. If messages span message queue records, the size of the input segment has no limit. When messages do not span message queue records, the maximum allowable size for input segments is *size2* minus the amount for various prefix items included with the data portion of the message queue record. The maximum size allowed for output segments is *size2* minus the amount for various prefix items included with the data portion of the LRECL (message queue record). Prefix items and sizes are displayed in Table 22. The Prefix is listed, and sizes for IMS Versions 7, 8, and 9 are provided with usage comments:

Table 22. Message Prefix Size

Prefix Section	IMS V7 Size (in bytes)	IMS V8 Size (in bytes)	IMS V9 Size (in bytes)	Comments
Basic <sup>1</sup>	64	64	64	All messages
System Segment (81)	56	56	56	All messages
	64	64	64	All messages with MFS
ISC/LU6.1 Prefix (84) <sup>3</sup>	22	22	22	All ISC/LU6.1 messages
Extended Prefix (86) <sup>4</sup>	16	16	16	All messages

Table 22. Message Prefix Size (continued)

Prefix Section	IMS V7 Size (in bytes)	IMS V8 Size (in bytes)	IMS V9 Size (in bytes)	Comments
APPC (LU6.2) (87)	128-512	128-512	128-512	All APPC messages
OTMA (87)	128-4096	128-4096	128-4096	All OTMA messages
Security Prefix (88) <sup>2</sup>	20	22	22	All messages if RACF or ETO is defined
Workload Manager (89)	16	24	24	All messages
System Extension (8A)	24	24	24	All messages
MSC Extension (8B)	92	104	104	All messages if MSC is defined
TMR (8C)	116	144	144	All messages
	102	144	144	All messages with shared queues
Conversation Ext. (8D) <sup>3,4</sup>	40	40	40	All conversational messages
TM/MSC User Prefix (8E) <sup>6</sup>	5-512	5-512	5-512	Dependent on using DFSMSCE0 User Prefix exit routine
IMS Internal Prefix (8F) <sup>7</sup>	5-8	5-512	5-512	Reserved for IMS
User Data Segments <sup>5</sup>	V	V	V	All messages

**Notes:**

1. All messages begin with a basic prefix segment that has no prefix code. If the message is a first or only queue buffer message, it has a basic prefix and some or all of the prefixes type 81 through 8D. User data segments can follow the IMS prefixes.

Prefix segments 81 through 8D begin with a 3-byte LLC field where LL is the 2-byte length of the prefix segment (including the LL field) and C is the 1-byte prefix code.

2. The security prefix is included in only the first message buffer if RACF or ETO is used.
3. Because they are not system definition values, the ISC/LU6.1 (type 84) prefix and the conversation extension (type 8D) prefix are always included in the maximum length of the IMS prefix calculated at IMS initialization. If Shared Queues is used, an additional 40 bytes is added.
4. All segments following the extended prefix header (type 86) are extended prefix segments. The IMS release level determines the size of the extended prefix segments created for a message. When MSC or Shared Queues are used, a message can be processed by an IMS that is of a different release level than the originating IMS. If the extended prefix segments lengths are greater for the target IMS, IMS can create or increase the size of the extended prefix segments. Extended prefix segments are never shortened.

**Recommendation:** In an environment where MSC and Shared Queues are used and IMSs are of different release levels, use the highest calculated values for *size1* and *size2* for each IMS.

5. User data segments follow the IMS prefix segments.
6. You can request that a user prefix be inserted into the message prefix. This is done with the MSC Routing exit routine (DFSMSCE0). The size of the user prefix can be from 5 to 512 bytes long.
7. The Workload Router Tool (product number 5697-B87) or another IMS tool can request that an internal prefix be inserted into the message prefix. The size of this user prefix can be from 5 to 512 bytes long.

For messages that span message queue records, the first item (basic) is included in each record of the message. Also, if RACF is generated, the RACF item plus a small (4-byte) system segment is included. All other prefix items, if applicable, appear only in the first record of the message.

The values for *size1* and *size2* should be tuned for your system.

**Related Reading:** For information on how to tune the values of *size1* and *size2* after IMS is operational, refer to “Message Queue Data Set Allocation” in *IMS Version 9: Installation Volume 1: Installation Verification*, and to “Initial Optimizing of IMS Buffer Pools” in *IMS Version 9: Administration Guide: System*.

The result of this tuning should yield a value for *size1* such that the I/O activity is equally split between the IMS.SHMSG data set, containing input and output message segments up to the length of *size1*, and the IMS.LGMSG data set, containing all input and output message segments larger than *size1*.

RECLNG can be changed prior to a cold start of IMS, but if a smaller value is specified prior to a BLDQ restart, the restart may abnormally terminate.

If emergency restart procedures using BUILDQ are to be used, reallocation of logical record and data set spaces must be done carefully. The BUILDQ procedure always restores the message queue entries to the relative position in the respective queue data sets they had at the time they were saved. If the logical record or data set size has been decreased, it may be impossible to perform the restart.

**Related Reading:** Refer to the information about restarting IMS in the *IMS Version 9: Operations Guide* for additional details.

#### SHUTDOWN=

Specifies the number of records to be reserved in each data set to allow the system to automatically shut down if the data set becomes filled with unprocessed messages. The maximum valid specification is 32767. The default is 30. In an XRF system, this parameter also reserves the number of records in each of the local message queue data sets (IMS.LGMSG, IMS.SHMSG, or IMS.QBLKSL) for shutdown.

The number of records specified should provide enough space in each data set to allow for an orderly shutdown, which depends on message throughput and the number of regions that are scheduled. In general, the following calculation applies:

Maximum number of output messages between synchronization points per application  $\times$  number of regions scheduled + 1 input message per logical terminal active at the same time.

**Example:** An example of the MSGQUEUE macro statement with all data sets on a 3380 is:

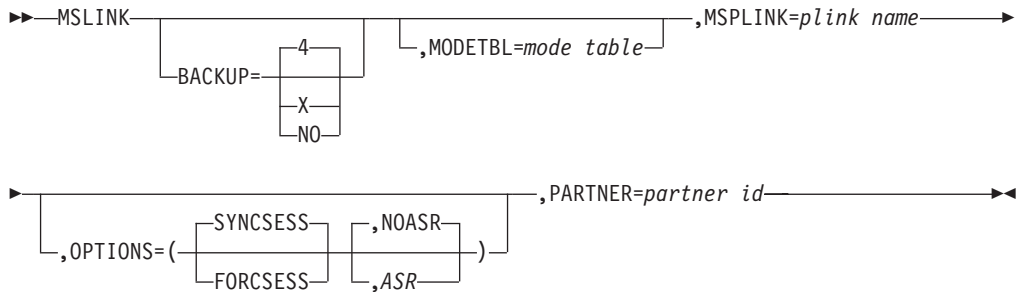
```
MSGQUEUE DSETS=(LGDK),BUFFERS=(40,2520),SHUTDOWN=200
```

This example indicates that the default sizes for the short and long message queue data sets are used. The size of each buffer block is 2520, allowing 1 long message record, 5 short message records or 45 queue block records to fit into a block.

For non-Shared Queues, 40 buffers are allocated and 200 records in each data set are reserved to allow the system to automatically shut down. For Shared Queues, the minimum buffer value of 200 is used, and the SHUTDOWN value is ignored.

## MSLINK Macro

The MSLINK macro statement defines a logical link to another system. It can be followed by one or more macro statements that define the link name blocks (MSNAME) if any are to be defined for this logical link.



## Keyword Parameters

To find which parameters apply to your IMS configuration, refer to Table 15 on page 33.

### BACKUP=

Allows (for XRF only) the users to override switching options specified on the MSPLINK macro. It controls the automatic BTAM or VTAM Multiple Systems Coupling (MSC) link restart after takeover. Use only if HSB=YES is specified on the IMSCTRL macro.

X is a numeric integer from 1 to 7, inclusive. It sets priorities for reestablishing the session. When either the keyword or the parameter is omitted, the default is 4. NO suppresses the session recovery of the MSC physical link at takeover.

Although requests to VTAM are prioritized by IMS, the active requests can be completed in any order because of internal VTAM conflicts and pacing.

### MODETBL=

Specifies the name of the VTAM logon mode table entry (logon mode name) containing the SNA bind parameters to be used when a session is established for this terminal. The maximum is 255 unique names for each IMS system.

This function allows a system definition specification for referencing an entry other than the default entry in the user's VTAM logon mode table.

With this function, if MODETBL= is not specified at system definition, no functional or operational change affects the user.

If MODETBL= is specified at system definition, the specified entry name is used.

The /DISPLAY command displays the current MODETBL name. The MODETBL name can be overridden by:

- VARY ACT, LOGON= command by the network terminal operator
- /RST or /CHANGE command by the master terminal operator

If the VTAM-node=name being defined is in another domain (that is, a cross-domain resource), the MODETBL parameter need not be specified.

**Related Reading:** See *Advanced Communications Function for VTAM (ACF/VTAM) Macro Language Guide* for additional information.

#### **MSPLINK=**

This operand identifies the physical link to which this logical link is assigned. If this operand is not specified, no assignment to a physical link is made and the /MSASSIGN command must be used to assign the logical link to a physical link before communication can be established between the two systems.

When specified, the MSPLINK macro must be defined before the MSLINK macro.

#### **OPTIONS=**

This parameter allows you to specify whether session initiation should be completed. SYNCSESS indicates that session initiation is to be completed only if session resynchronization is successful. FORCSESS forces the session to be completed regardless of whether session resynchronization is successful. Successful session resynchronization occurs when the message sequence numbers of the two logical units in session agree, or when the sequence number of the sender is no more than 1 higher than the sequence number of the receiver. The default is SYNCSESS. The specifications made on this OPTIONS parameter must be consistent with the MSLINK OPTIONS specification in the partner systems. During IMS execution, this option can be overridden by the /CHANGE command.

The OPTIONS parameter is valid for MSC VTAM links only.

When using the Session Outage Notification facility, the second parameter allows you to specify automatic session restart (ASR) processing for a defined node. The default is NOASR.

Specifying ASR or NOASR on the MSLINK macro overrides ASR specifications on the TYPE and MSPLINK macros.

Use the /DISPLAY command to display the current ASR option, and the /CHANGE command to change it.

ASR works only if both sides of the link are using the ASR option.

#### **PARTNER=**

The partner-id is a two-character alphabetic identification. It ensures that the two related logical links in two systems are always logically and physically connected. Both systems must have MSLINK macro statements with the same partner-id. A logical link can be assigned to a different physical link; the two systems still communicate via the logical link, which remains as defined.

---

## **MSNAME Macro**

The MSNAME macro statement provides a name for the remote and local system identifications that it represents. When used, it can be followed by one or more NAME macros defining remote logical terminals.

▶▶—MSNAME—SYSID=(*remote system id,local system id*)————▶▶

## Label Field

*linkname* specifies a one-to eight- character alphanumeric name used externally in commands. Linknames, block names, transaction codes, and logical terminal names, collectively, cannot contain duplicates. Specifying the *linkname* is required.

## Keyword Parameters

To find which parameters apply to your IMS configuration refer to Table 15 on page 33.

### **SYSID=**

*remote system identification* identifies the remote system that is represented by this name.

The *local system identification* specifies the system identification that is to be used for routing messages back to this system.

*remote system identification* specification must not have been previously defined as:

- A remote system identification within a previous MSNAME macro statement.
- A local system identification within an APPLCTN, TRANSACT, or MSNAME macro statement.

*local system identification* specification must not have been previously defined as a *remote system identification* within a previous MSNAME macro statement.

Values from 1 through 2036 are valid. A one-to-one relationship exists between remote SYSIDs and MSNAMEs. APPLCTN and TRANSACT macro statements must refer to remote SYSIDs specified in MSNAME statements in the same system definition when defining remote applications.

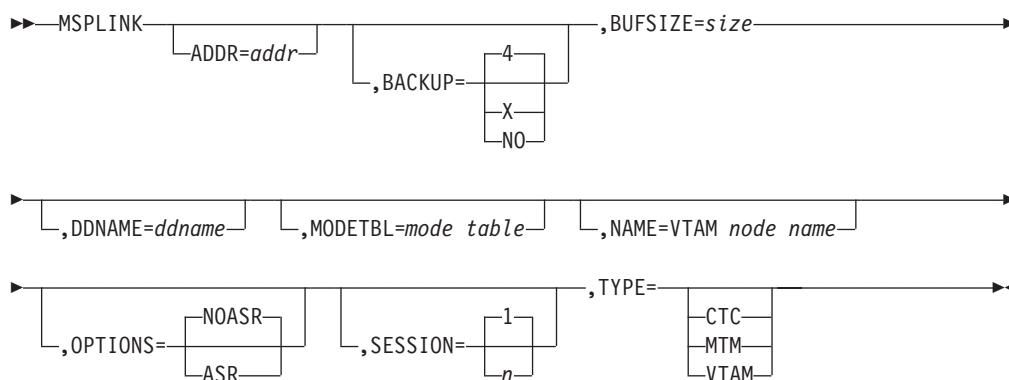
---

## MSPLINK Macro

The MSPLINK macro statement defines a physical link. It can define three different types of connections between two systems: channel-to-channel connection (CTC), main-storage to main-storage connection (MTM), and ACF/VTAM.

Logical links are assigned to physical links during system definition (MSPLINK operand in MSLINK macro) or dynamically with /MSASSIGN commands.

**Related Reading:** Refer to *IMS Version 9: Command Reference* for detailed information on the /MSASSIGN command.



## Label Field

The MSPLINK name specifies a one- to eight-character alphanumeric name for this physical link. You must specify this name. This name is used (in the MSPLINK=keyword operand of the MSLINK macro) to assign a logical link to this physical link.

## Keyword Parameters

To find which parameters apply to your IMS configuration, refer to Table 15 on page 33.

### ADDR=

Specifies the address of the communication line or the address of the channel-to-channel adapter. The address value can be three or four hexadecimal digits. This operand is required for CTC link types and invalid for MTM and VTAM link types.

All devices attached to the same channel as the CTC adapter must be accessible through an alternate channel. In addition, system resources (for example, paging devices) should not be attached to the same channel as the CTC adapter.

### BACKUP=

Specifies (for XRF only) VTAM MSC link restart after takeover. Use only when HSB=YES is specified on the IMSCTRL macro.

`X` is a numeric integer from 1 to 7, inclusive, that specifies the priority for reestablishing the session. The default is 4 when either the keyword or the parameter is omitted. Specifying `BACKUP=NO` suppresses the automatic restart of the MSC physical link.

Although IMS prioritizes requests to VTAM, the active requests can be completed in any order because of internal VTAM conflicts and pacing.

### BUFSIZE=

Specifies the input and output buffer sizes for each logical link defined for this physical link. For example, if three logical links are defined under a physical link specifying `BUFSIZE=nn`, six buffers each equal to `nn` (three input and three output) are allocated. The systems on each end of the physical link must specify equal buffer size. Consistency is checked during link startup. Minimum size is 160. For performance considerations, the specified buffer size should be 160 plus the larger of the maximum SPA size or maximum segment size.



MSC VTAM links require a minimum buffer size of 208 bytes and allow a maximum size of 30720 bytes.

LU 6.2 application transactions transmitted over MSC links need larger link buffers to accommodate the extra 500 bytes for the LU 6.2 message prefix and the IMS extended prefix. The buffer size must be a decimal value between 208 and 30720 that can be expressed by the algorithm  $X \text{ times } 2 \text{ to the power of } Y$ .  $X$  must be a value from 8 through 15, and  $Y$  must be a value from 3 through 11. This is a VTAM restriction. Thus, for example, the value 144 (representing  $9 \times 2^4$ ) and the value 28672 (representing  $14 \times 2^{11}$ ) are acceptable values.

**Related Reading:** Details of this VTAM restriction are found in *ACF/VTAM Macro Language Reference*.

This restricts the valid MSC VTAM buffer sizes to one of the following decimal values:

112	448	1792	7168	28672
120	480	1920	7680	30720
128	512	2048	8192	
144	576	2304	9216	
160	640	2560	10240	
176	704	2816	11264	
192	768	3072	12288	
208	832	3328	13312	
224	896	3584	14336	
240	960	3840	15360	
256	1024	4096	16384	
288	1152	4608	18432	
320	1280	5120	20480	
352	1408	5632	22528	
384	1536	6144	24576	
416	1664	6656	26624	

The value specified plus the size of any routing information later appended by VTAM must be less than the maximum message size specified for VTAM and its related NCPs.

#### **DDNAME=**

This is the ddname of the JCL statement that describes this physical connection. This operand is required CTC link types and is invalid for MTM and VTAM link types.

Each name must begin with an alphanumeric character. The following names cannot be used as ddnames:

DFSRESLB	IMSLOG	IMSTFMT	MSDBINIT
DUMP	IMSLOGR	IMSUDUMP	PRINTDD
IEFRDER	IMSLOGR2	LGMSG	PROCLIB
IEFRDER2	IMSLOG2	MATRIX	QBLKS
IMSACB	IMSMON	MSDBCP1	SHMSG
IMSDBL	IMSRDS	MSDBCP2	
IMSDILIB	IMSSPA	MSDBDUMP	

#### **MODETBL=**

Specifies the name of the VTAM logon mode table entry (logon mode name) containing the SNA bind parameters to be used when a session is established for this terminal.

MODETBL= allows a system definition specification for referencing an entry other than the default entry in the user's VTAM logon mode table. Normally, the terminal operator specified this mode table entry name when logging on to a terminal; this was not possible, however, from the IMS master terminal because IMS initiated the session.

If MODETBL= is not specified at system definition, no functional or operational change affects the user.

If MODETBL= is specified at system definition, the specified entry name is used.

The /DISPLAY command displays the current MODETBL name. The MODETBL name can be overridden by:

- The LOGON APPLID entry by the remote terminal operator
- VARY ACT, LOGON= command by the network terminal operator
- /OPNDST, /RST, or /CHANGE command by the master terminal operator

The MODETBL= parameter is required for the IMS master terminal if the VTAM default mode table has not been configured specifically for the device to be used as the IMS master terminal.

Specifying MODETBL on the MSPLINK macro circumvents the need to specify the mode table entry when logging on to terminals that always require specification of the same mode table entry name.

If the terminal being defined is in another domain (that is, a cross-domain resource), the MODETBL parameter need not be specified.

**Related Reading:** See *Advanced Communications Function for VTAM (ACF/VTAM) Macro Language Guide* for additional information.

#### **NAME=**

Specifies the VTAM node name of the remote system at the other end of the link. NAME= must be the same as the label on the VTAM APPL statement for the remote system (that is, the minor node name). For an MSC VTAM link communicating with an XRF complex, the node name must be the VTAM USERVAR associated with the partner IMS/XRF complex. If the ACBNAME parameter of the VTAM APPL statement is not specified, NAME= is the same as the application identification (APPLID=) specified on the IMS COMM macro statement in the remote system. This operand is required for MSC VTAM links only and is invalid for other link types.

#### **OPTIONS=**

The Session Outage Notification facility only, specifies automatic session restart on all logical links associated with physical links. The default is NOASR. ASR or NOASR in the MSLINK macro overrides ASR definitions on the MSLPINK macro.

Use the /DISPLAY command to display the current ASR option, and the /CHANGE command to change it.

ASR works only if both sides of the link are using the ASR option.

#### **SESSION=**

Specifies the number (from 1-676) of parallel sessions that can be active for the physical link defined by the MSPLINK macro statement. It is valid for MSC VTAM links only. The default is 1.

A large SESSION value enables the user to dynamically assign more logical links to the physical link than were originally assigned during system definition. The SESSION value can be increased at system startup by using JCL. For instructions, see the keyword NLXB described in Chapter 4, "Procedures," on page 191.

Because a given number of logical parallel sessions uses the same amount of storage for control blocks and buffers as would the same number of physical links, allocating many parallel sessions can use an excessive amount of common storage area. You should predetermine how much common storage area you want to use.

**TYPE=**

Defines the type of physical link being described. Valid parameters are CTC, MTM, and VTAM.

The MTM type link allows more than one IMS system to run in the same OS/390 system without a hardware link. It is intended for backup or testing.

---

**NAME Macro**

The NAME macro statement defines a logical terminal name (LTERM) associated with a physical terminal. Preparation of the NAME macro can be required for each of the macro sets that you prepare to define your IMS system.

The logical terminal can be specified as the IMS master terminal if the associated physical terminal is a 3270 display terminal, SLU 1, or SLU 2. The terminal must be nonswitched. If the terminal is a SLU 1, both the input (ICOMPT) and the output (COMPT) component designations must refer to the first physical component. The first physical component of a SLU 1 terminal must be a console if it is to be defined as the IMS master terminal.

If the IMS master terminal is specified as a 2740 or SLU 1, the designation of a secondary master terminal is optional and can be specified if the logical terminal's associated physical terminal 2740 (but not an NTO device), or SLU 1. The terminal must be nonswitched. If the terminal is a SLU 1, both the input (ICOMPT) and the output (COMPT) component designations must refer to the first physical component. The first physical component of a SLU 1 can be either a console or a printer if it is to be defined as the IMS secondary master terminal.

If the IMS master terminal is specified as a 3270 display terminal, a secondary master terminal is required, and must be specified for a 328x, associated physical terminal.

If the IMS master terminal is specified as SLU 2, you must specify a secondary master terminal in a preceding NAME macro for a 2740, 328x, SLUTYPE1, or SPOOL associated physical terminal.

If other physical terminals are defined on the same communication line as the primary or secondary master terminal, then special restrictions will apply to the use of the /CHECKPOINT and /IDLE commands in the IMS system.

**Related Reading:** Details of these restrictions are contained in the corresponding command descriptions in *IMS Version 9: Command Reference*.

The Message Format Service (MFS) special master terminal formatting is only available for 3270 display terminals specified with a screen size of 24 x 80.

For output purposes, the COMPT operand associates the specified logical terminal with a specific component in a 2780, 3275, 3601, SLU 1, or SLU P terminals, or LU 6.1 nodes. The COMPT operand is valid only for these terminal types.

For SLU 1 terminals, the NAME macro statement also defines the data set at the terminal to receive output for corresponding components defined in the TERMINAL macro statement as USERDS1.

For LU 6.1 nodes, the position of the NAME macro within the system definition macro set determines whether LTERM allocation is fixed or dynamic.

If the allocation of LU 6.1 logical terminals is to be predefined (fixed), the NAME macro statement must follow the TYPE and TERMINAL macro statements, and the SESSION parameter must be equal to 1 (by specification or default).

If the allocation of LU 6.1 logical terminals is to be dynamic, the NAME macro statement must follow the SUBPOOL macro statement, and the SESSION parameter must specify a valid value in the range 1 through 255. The default is 1.

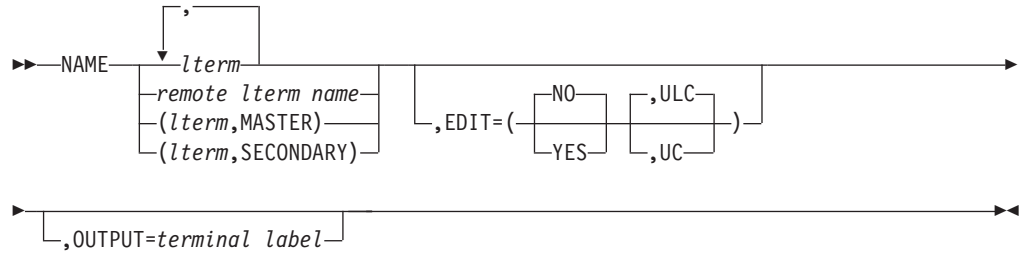
If two or more sessions are defined on the SESSION parameter, dynamic allocation must be used.

It can be desirable to use dynamic rather than fixed allocation for one parallel session (SESSION=1). This permits all subpools to be available for dynamic allocation to that session. This function is similar to SUBPOOL support offered for BTAM dial terminals.

Figure 14 summarizes the order of the macros:

TYPE	UNITYPE=LUTYPE6TYPE	UNITYPE=LUTYPE6
TERMINAL	SESSION=1	TERMINAL SESSION=(1-255)
NAME	ltrmname	.
		.
		VTAMPOOL
		SUBPOOL NAME=SBPLNAME
		NAME ltrmname

*Figure 14. Order of Macro Statements for Fixed and Dynamically Allocated Subpools*



**Master Terminal**



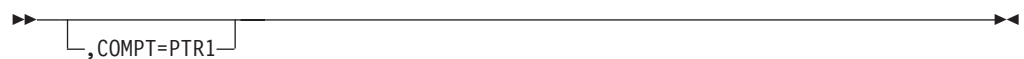
**3601 Work Station**



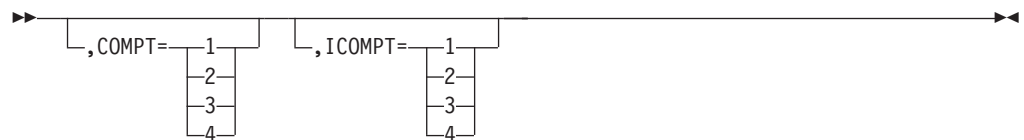
**2780 Terminal**



**3275 Terminal**



**SLUTYPE1, SLUTYPEP, LUTYPE6 Terminals**



**Positional Parameters**

**lterm or lterms**

Specifies one or more 1 to 8-character names for logical terminals associated

with previously defined physical terminals. The name can contain only alphanumeric characters, that is, the letters A through Z, digits 0 through 9, and national characters # \$, and @. WTOR and DFSMTCNT are invalid *lterm* names. The entered name cannot begin with the character sequence INQU. Logical terminal names, transaction codes, and MSNAME link names, collectively, cannot contain duplicates. This operand is required. For further restrictions, refer to "Resource Naming Rules" on page 60.

If you enter the operand as a sublist with the keyword MASTER as a second parameter, the entered name is the identifier of the IMS master terminal.

If you enter the operand as a sublist with the keyword SECONDARY as the second parameter, the entered name is the identifier of the IMS secondary master terminal.

Examples of multiple specifications are:

```
NAME LTERMA,LTERMX,LTERMZ
NAME LTERMA,(LTERMX,MASTER),LTERMZ
```

One logical terminal in the system must be specified as the master terminal. The logical terminal chosen as the master terminal cannot be on a switched LINEGRP. It must be one of the following types:

- 2740 with station control
- 3270 display terminal
- Terminal defined as either SLUTYPE1 or SLUTYPE2

It must also be the console component if the terminal is defined as SLUTYPE1.

One logical terminal can be specified as the secondary master terminal. This logical terminal cannot be on a switched line group. It must be one of the following types:

- A 2740 with station control
- A 328x physical terminal
- A terminal defined as SLUTYPE1
- A terminal in a line group defined as a SPOOL

If a 2740 is chosen as the secondary master terminal, the logical terminal chosen as the primary master terminal and the logical terminal chosen as the secondary master terminal must be associated with the same 2740 physical terminal.

For terminals with multiple NAME LTERMs assigned, system error message responses are sent to the logical terminal name that is lowest in the collating sequence, unless the terminal is an input-only terminal. For input-only terminals, the system responses go to the first assigned logical terminal OUTPUT= terminal.

If multiple NAME LTERMs are assigned and an LTERM other than the first is specified as MASTER or SECONDARY, data and commands are not designated by the system as having originated at a master terminal.

#### **remote lterm name**

Specifies a one- to eight-character name for a logical terminal associated with a physical terminal defined in a remote IMS system. The name can contain only alphanumeric characters. WTOR and DFSMTCNT are invalid logical terminal names. The name cannot begin with the character sequence INQU. For further

restrictions, refer to "Resource Naming Rules" on page 60. No other operands are meaningful for a remote logical terminal; if provided, they are interpreted as comments only.

## Keyword Parameters

To find which parameters apply to your IMS configuration refer to Table 15 on page 33.

### COMPT=

Specifies the output component associated with this logical terminal. If this is the master terminal, the COMPT= operand value must be either not specified or 1.

For 3601 work stations, you can specify 1, 2, 3, or 4.

For 2780 terminal components, you can specify 1 or PTR1, or 2 or PCH1.

For 3275 terminals, you can specify only PTR1. PTR1 refers to a 3284 Model 3 printer attached to a 3275 terminal, within a remote 3270 line group or switched 3275.

COMPT= cannot be specified for NTO devices.

### EDIT=

Specifies whether the logical terminal user-supplied edit routine DFSCNTE0 is to be used when routing a message to this logical terminal. The logical terminal edit routine is only included within the IMS control program nucleus if one of the NAME statements specifies EDIT=YES. If ULC is specified, output is transmitted as received. If UC is specified, output is translated to uppercase before transmission.

The logical terminal edit routine DFSCNTE0 is not supported for ETO logical terminals.

### ICOMPT=

Specifies the input component associated with the terminal defined in the previous TERMINAL macro. If this is the master terminal, the ICOMPT= operand value must be either not specified or 1.

For SLU 1, SLU P, or LU 6.1 terminal systems, you can specify 1, 2, 3, or 4. These values relate to the components specified in the preceding TERMINAL macro statement.

For secondary logical units defined as SLUTYPE1, if the first component is a console, the default is 1; if the first component is a printer, no default exists. For terminals defined as SLUTYPEP or LUTYPE6, the default is 1.

ICOMPT= cannot be specified for NTO devices.

### OUTPUT=

Specifies a terminal that is to be used as the output terminal for this logical terminal name. This is specified by entering the label of the TERMINAL macro defining the desired terminal. The referenced TERMINAL statement must occur before this NAME statement. OUTPUT= cannot be specified for VTAM terminals. OUTPUT= cannot be used with response mode, Fast Path, or conversational transactions.

A split input/output LTERM is not applicable to ACF/VTAM terminals.

## POOL Macro

The POOL macro statement describes a pool of logical terminals that are to be associated with a set of switched communication lines. The IMS user needs only one logical terminal pool for all switched communication lines. All POOL macro instructions must follow all LINE macro instructions within a switched line group.

▶▶—POOL—◀◀

**Related Reading:** To find if this macro applies to your IMS configuration, refer to [Table 15 on page 33](#). Refer to “Switched Communication Network” in the “Logical Terminal Network Design” of *IMS Version 9: Administration Guide: System* for a discussion on the use of pools and subpools.

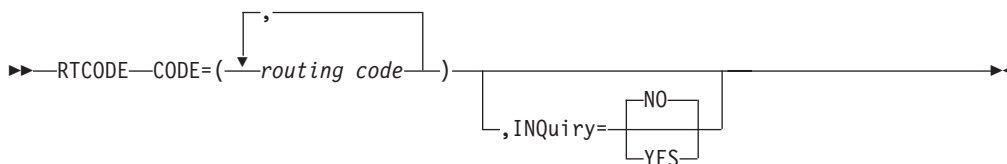
## RTCODE Macro

The RTCODE macro statement is used one or more times with the APPLCTN macro statement that defines an IMS Fast Path application program. It specifies the routing codes that identify the program named in the preceding APPLCTN macro statement. A TRANSACT macro statement that specifies an IMS Fast Path—exclusive transaction generates an internal RTCODE macro statement with a routing code identical to the transaction code.

A RTCODE must not be a duplicate of a Fast Path—exclusive transaction code. Duplicates can be identified by using the NAMECHK option of the IMSCTRL macro.

The edit/routing exit routine must be used to route the transaction to the correct Fast Path application program.

Routing codes can be added, changed, or deleted by a MODBLKS system definition. However, new routing codes can only be added if the online system to which they are to be added already has Fast Path defined for it.



## Keyword Parameters

To find which parameters apply to your IMS configuration refer to [Table 15 on page 33](#).

### CODE=

Specifies the one- to eight-character alphanumeric routing code, or list of routing codes. The first character must be either a letter or a digit. Routing codes can be duplicates of either transaction codes or logical terminal names, but each must be unique within the set of routing codes.

### INQUIRY=

Specifies whether (YES) or not (NO) any message associated with the routing code specified on the same RTCODE macro statement is an inquiry transaction. The default is NO. INQ=YES should be specified only for



transactions that do not cause a change to a database. Programs are prohibited from issuing Insert, Delete, or Replace calls to a database when processing a transaction defined as INQ=YES.

An APPLCTN macro statement that defines a message-driven Fast Path application program should be followed by at least a TRANSACT macro statement or a RTCODE statement.

---

## SECURITY Macro

The SECURITY macro statement lets you specify optional security features to be in effect during IMS execution unless they are overridden during system initialization.

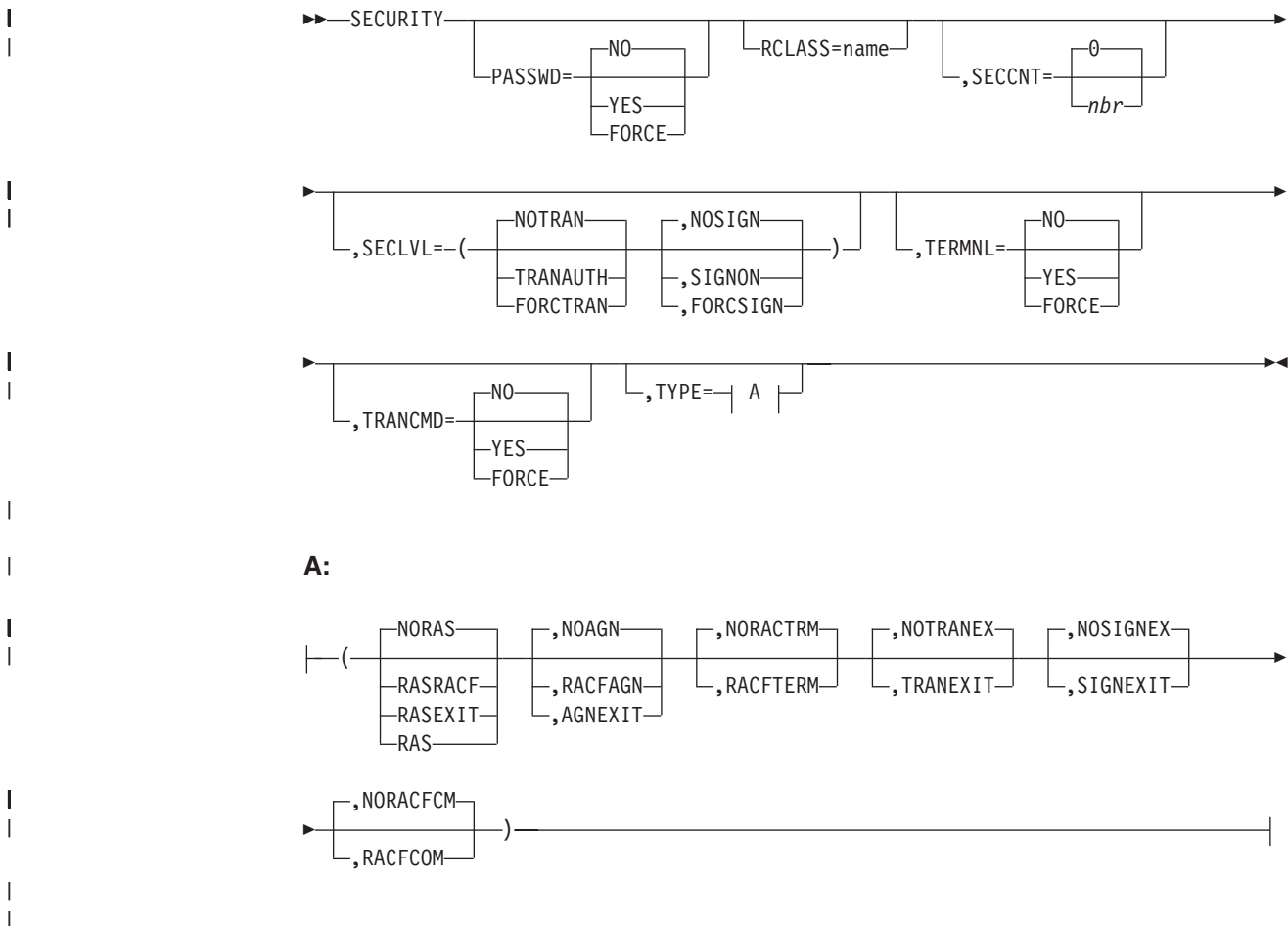
The IMS system can be defined to use the Resource Access Control Facility (RACF) licensed program (or equivalent), an exit routine, or both to perform the following types of security protection:

- Transaction authorization
- Command authorization
- Signon verification
- PSB authorization verification
- Application resource, or DBCTL, access authority checking

Authority to issue specific IMS commands can be restricted to certain transactions, and password and terminal security requirements can be imposed.

**Related Reading:** For full details of the security options and their use, see “Establishing IMS Security” in *IMS Version 9: Administration Guide: System*.

If the SECURITY macro is included, any values specified or accepted by default override options specified in the COMM or MSGEN macro.



## Relationships between the SECURITY, COMM, and IMSGEN Macros

The SECURITY macro is provided to enhance IMS system security. It can replace the security specifications on the IMSGEN and COMM macros, both explained elsewhere in this manual. This section describes the relationship that exists between these three macros and suggests a possible installation strategy.

All security specifications have now been consolidated in the SECURITY macro. However, for compatibility with previous releases, the COMM or IMSGEN keywords related to security specifications can still be used.

The security-related keywords from these three macros are accepted hierarchically in the order SECURITY, COMM, and then IMSGEN; that is, if the SECURITY macro is used, the specifications and defaults from that macro will take precedence over any security specifications coded on either COMM or IMSGEN.

If security specifications are coded on more than one of these macros, a warning message is issued, indicating that the specifications and defaults from the highest macro in the hierarchy present in the stage 1 input is used. Likewise, if the COMM macro is used and any of the IMSGEN macro's general communication options are specified, a warning message is issued indicating that the general communication specifications and defaults from the COMM macro are used.

## Keyword Parameters

To find which parameters apply to your IMS configuration, for example to DBCTL, refer to Table 15 on page 33.

### **PASSWD=**

Specifies whether or not the password security specifications established by the Security Maintenance utility will be in effect at the completion of a normal restart.

#### **NO | YES | FORCE**

Specifies whether (FORCE or YES) or not (NO) password security tables are to be loaded from the MATRIX data set during IMS restart.

If YES or NO is specified, this option can be overridden by the MTO during restart. The default is NO.

### **RCLASS=name**

Specifies an identifier of 1 to 7 alphanumeric characters that is to be used to identify the IMS system as a resource class to RACF for transaction authorization and user ID verification.

The specification is valid only if one of the following is true:

- TYPE=RACFTERM is specified on the system definition SECURITY macro; or RCF=A,T, or Y is specified on the IMS procedure.
- TYPE=RACFAGN, RASRACF, or RAS is specified on the system definition SECURITY macro, or ISIS=1, R, or A is specified on the IMS procedure.

IMS class names do not need to be unique. If you do make them unique, you can, for example, define the same transaction name on a production subsystem and a test subsystem, with different access lists for each subsystem, with no ambiguity.

The RCLASS= specification on the DFSDCxxx member overrides the RCLASS= specification on the system definition SECURITY macro.

### **SECCNT=**

Specifies the maximum number of terminal and password security violations to be accepted per physical terminal and the number of transaction command violations per transaction prior to master terminal notification of such violations. The default is 0, which nullifies notification to the master terminal. The number specified must be 0, 1, 2, or 3.

If SECCNT is not 0, the master terminal is notified for every violation.

### **SECLVL=**

Specifies the user ID verification and transaction authorization security enforcement level options available to the MTO when restarting IMS.

**Related Reading:** For further information on the effect of these options and the relationship between these operands, JCL, and the /NRE command, refer to *IMS Version 9: Administration Guide: System*.

#### **NOTRAN | TRANAUTH | FORCTRAN**

Specifies whether (FORCTRAN or TRANAUTH) or not (NOTRAN) transaction authorization is to be performed by the online IMS system.

If TRANAUTH or NOTRAN is specified, this option can be overridden by the MTO at restart. The default is NOTRAN.

A specification of FORCTRAN or TRANAUTH requires that TYPE=RACFTERM or TRANEXIT, so that the processing can be performed.

If FORCTRAN or TRANAUTH is specified, the requirements for user ID verification must also be considered.

**Related Reading:** For further details on the effect of these options, refer to *IMS Version 9: Administration Guide: System*.

### **NOSIGN | SIGNON | FORCSIGN**

Specifies whether (FORCSIGN or SIGNON) or not (NOSIGN) /SIGN ON user ID verification security tables are to be loaded from the MATRIX data set during IMS restart.

If SIGNON or NOSIGN is specified, this option can be overridden by the MTO at restart. Specification of FORCSIGN or SIGNON requires that TYPE=RACFTERM or SIGNEXIT, so that processing can be performed.

Transaction authorization requires a validated user ID in order to verify that the user is authorized to invoke the transaction. For this reason, the security level of user ID verification must be at least as great as that of transaction authorization. The requirements are shown in the following table:

Parameter	Default	Invalid
NOTRAN <sup>4</sup>	NOSIGN	- -
TRANAUTH	SIGNON	NOSIGN
FORCTRAN	FORCSIGN	SIGNON
FORCTRAN	FORCSIGN	NOSIGN

### **TERMNL=**

Specifies whether or not the terminal security specifications established by the Security Maintenance utility will be in effect at the completion of a normal restart.

### **NO | YES | FORCE**

Specifies whether (FORCE or YES) or not (NO) terminal security tables are to be loaded from the MATRIX data set during IMS restart.

If YES or NO is specified, this option can be overridden by the MTO during restart. The default is NO.

**Related Reading:** For further details on the effect of these options, refer to *IMS Version 9: Administration Guide: System*.

### **TRANCMD=**

Specifies whether or not the transaction command security specifications established by the Security Maintenance utility will be in effect at the completion of a normal restart.

### **NO | YES | FORCE**

Specifies whether (FORCE or YES) or not (NO) transaction command security tables (if any) are to be loaded from the MATRIX data set during IMS restart.

If YES or NO is specified, this option can be overridden by the MTO during restart by specifying the NOTRANCMD or TRANCMD keyword, respectively, on the /NRESTART command. The default is NO.

4. Specified or default.

**Related Reading:** For further details, refer to *IMS Version 9: Administration Guide: System*.

#### TYPE=

Specifies whether application group name authorization, application resource access authorization, transaction authorization, or signon user ID verification is to be performed.

#### **NORAS** | **RASRACF** | **RASEXIT** | **RAS**

Specifies whether (RASRACF, RASEXIT, RAS) or not (NORAS) application resource access security authorization is to be performed by IMS at execution time. Resource access security checking can be performed by the RACF licensed program (RASRACF), by an exit routine named DFSRAS00 (RASEXIT), or by both RACF and an exit routine (RAS). The default is NORAS.

**Restriction:** You cannot use both application group name security and resource access security. When both types of security are specified on the TYPE keyword, resource access security is used.

#### **NOAGN** | **RACFAGN** | **AGNEXIT**

Specifies whether (RACFAGN or AGNEXIT) or not (NOAGN) application group name authorization is to be performed by IMS at execution time. Application group name security checking can be performed by the RACF licensed program (RACFAGN) or by an exit routine named DFSISIS0 (AGNEXIT). The default is NOAGN.

**Recommendation:** If you are currently using AGN security to protect the IMS resources used by IMS dependent regions, you might want to migrate your current AGN SMU-based security implementation to a RAS security implementation based upon RACF, or an equivalent product. SMU security is not supported for future releases of IMS. Any new security implementations should use the RAS parameters available on the TYPE keyword.

**Related Reading:** For more information about migrating SMU security implementations to RACF, or an equivalent product, see *IMS Version 9: Release Planning Guide*.

#### **NORACTRM** | **RACFTERM**

Specifies whether (RACFTERM) or not (NORACTRM) linkage to the RACF program product is to be provided for transaction authorization or /SIGN ON user ID verification. If RACFTERM is specified, RACF is invoked whenever either of the two functions is required. The default is NORACTRM.

#### **NOTRANEX** | **TRANEXIT**

Specifies whether (TRANEXIT) or not (NOTRANEX) the user transaction authorization exit (DFSCTRN0) will be called by IMS to process transaction authorization requests at execution time. If both RACFTERM and TRANEXIT are specified, RACF is called first. The default is NOTRANEX.

#### **NOSIGNEX** | **SIGNEXIT**

Specifies whether (SIGNEXIT) or not (NOSIGNEX) a user /SIGN ON exit (DFSCSGN0) is to be called by IMS to validate a user ID when entered at execution time. If both RACFTERM and SIGNEXIT are specified, RACF is called first. If TYPE=SIGNEXIT is specified in the SECURITY macro, SECLVL=SIGNON is also required to enforce user ID verification.

The default is NOSIGNEX; however, SIGNEXIT is assumed and forced if TRANEXIT is specified and NORACTRM is specified or accepted by default, because user ID verification is a prerequisite for transaction authorization checking.

#### **NORACFCM | RACFCOM**

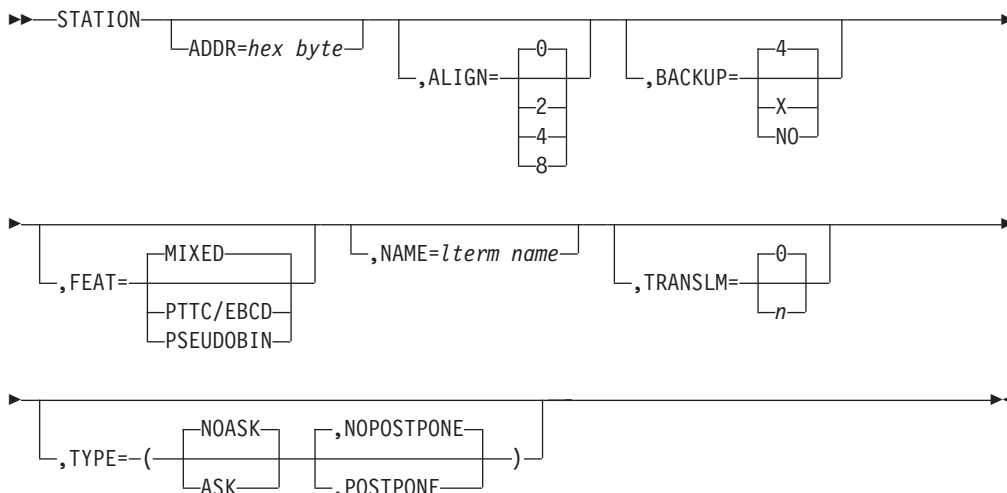
Specifies whether (RACFCOM) or not (NORACFCM) RACF is to be called to verify command authorization. This applies to commands from ETO terminals. The default is NORACFCM.

## SPAREA Macro

The SPAREA macro is no longer supported because DASD SPAs are no longer supported. If you specify the SPAREA macro, it is ignored.

## STATION Macro

The STATION macro statement describes the physical and logical characteristics of the System/3 or System/7.



## Keyword Parameters

To find which parameters apply to your IMS configuration refer to Table 15 on page 33.

#### **ADDR=**

Specifies the two-digit hexadecimal representation of the System/3 or System/7 address. This operand is not required for System/3 or System/7 contention line groups.

If either of the following conditions exists on the stage 1 input macros for a System/3 or a System/7 BSC line group, the address coded on the TERMINAL macros following the STATION macro must not be of the form XYXY, where X and Y are any hexadecimal digits:

- The address (ADDR=) is not coded on the STATION macro, and the coded address on the preceding LINE macro is of the form 0XY.
- The coded address on the STATION macro is of the form XY.

#### **ALIGN=**

Specifies that each data segment is to be right justified or padded to halfword,

fullword, or doubleword (2, 4, or 8, respectively) alignment. 0 specifies that no alignment is to be performed. The default is 0.

**BACKUP=**

Specifies (for XRF only) the control of the automatic terminal switching after takeover. Use only when HSB=YES is specified on the IMSCTRL macro.

X is a numeric integer from 1 to 7, inclusive, that sets priority for reestablishing the session. It defaults to 4 when either the keyword or the parameter is omitted. Specifying BACKUP=NO suppresses the session recovery of the terminal at takeover.

**FEAT=**

Specifies whether data blocks are to be transmitted in PTTC/EBCD (PTTC/EBCD) code, pseudo binary PTTC/EBCD (PSEUDOBIN) code, or either (MIXED). This operand is valid for System/7 (START/STOP) line groups only. The default is MIXED.

**NAME=**

Specifies a logical terminal name for the station CNT. The default is a system-assigned sequence number with the prefix RSTS.

**TRANSLM=**

Specifies the maximum number of blocks to be transmitted to the System/3 or System/7 for each output request. This operand must be specified as a decimal number from 0 through 15. A specification of 0 indicates no limit. The default is 0.

**TYPE=**

Specifies whether output should be allowed to queue until such time as the station asks for the output (ASK), or whether output should be sent to the station as it becomes available (NOASK). The default is NOASK.

The second parameter specifies that this station should not receive output (POSTPONE) or should receive output (NOPOSTPONE) when the station is started. The default is NOPOSTPONE.

**SUBPOOL Macro**

**For Switched Communication Devices:** The SUBPOOL macro statement, when used within a switched communication device macro set, defines a set of logical terminals. This set is within a pool of logical terminals. The set is associated with a physical terminal on a switched communication line when the /IAM command is executed. Subpools can be defined within a POOL macro statement. At least one subpool must be defined for each POOL macro statement.

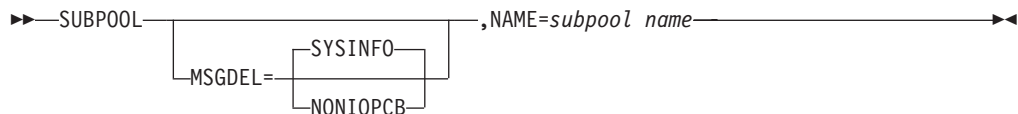
When used within a switched communication device macro set, the SUBPOOL macro statement has no operands. It must be followed by one or more NAME macro statements. These NAME statements can contain the COMPT= operand. If the COMPT= operand is present on any NAME statement within a subpool, it must be defined for a switched 3275. If the COMPT= operand is not present on any NAME statement within a SUBPOOL, it can be defined within a 2740 line group or for a switched 3275.

**Related Reading:** Refer to “Switched Communication Network” in the “Logical Terminal Network Design” in *IMS Version 9: Administration Guide: System* for a discussion on the use of pools and subpools.

**For LU 6.1 VTAM Devices:** The SUBPOOL macro statement, when used in a VTAM macro set, is a delimiter between groups of NAME macro statements to create LU 6.1 LTERM subpools. Although at least one NAME macro statement must be defined within the VTAMPOOL, it is valid to define one or more subpools with no NAME statements. This definition creates a reserved subpool to which no terminals are allocated until reassigned there by use of the /ASSIGN command.

The subpools must be named to be used in IMS commands or session initiation requests. The LU 6.1 subpool names, specified on the required NAME keyword operand of the SUBPOOL macro statement, must follow standard z/OS naming conventions and must be unique with respect to other LTERM subpool names within the VTAMPOOL. The subpool names must be unique among themselves, but need not be unique to other names used in an IMS system, such as LTERM names and transaction names. These subpools can only be used with statically defined ISC terminals. They cannot be used with ETO terminals.

If parallel sessions are defined in your system, at least one subpool must be defined. It is recommended that at least one subpool be defined for each parallel session defined in your system. Additional subpools can be defined if you desire. For a session to be brought up, a subpool must be allocated to it. Thus, if you have defined two parallel sessions in your system and only one subpool, only one session can be brought up at any one time.



## Keyword Parameters

To find which parameters apply to your IMS configuration refer to Table 15 on page 33.

### MSGDEL=

Used for VTAM LU 6.1 devices and specifies which message types IMS should discard for this terminal. The default is SYSINFO.

#### SYSINFO | NONIOPCB

SYSINFO, the default, specifies that IMS should discard DFS059 TERMINAL status messages for this terminal and also DFS3650 for SLU P devices.

NONIOPCB specifies that IMS should discard the following message types destined for this terminal:

- Message switches
- Messages inserted by an application program to an alternate PCB
- /BROADCAST messages
- DFS059 TERMINAL status messages

### NAME=

Used for VTAM LU 6.1 devices and names the LU 6 subpool. The NAME keyword parameter is not allowed for BTAM subpools.



---

## TERMINAL Macro

The TERMINAL macro statement defines physical and logical characteristics of VTAM nodes and non-VTAM communication terminals.

The NAME macro statements that follow a TERMINAL macro statement supply the logical terminal names that are associated with the physical terminal at system definition. Whichever terminal name is designated in the first NAME macro statement that follows a TERMINAL macro statement becomes the response or input/output logical terminal.

A TERMINAL macro statement that defines terminals connected to a switched communication line must not be immediately followed by a NAME macro statement.

All Fast Path-eligible terminals must operate in response (forced or transaction) mode. In addition, RESP=TERM must be specified on the LINE macro statement for Fast Path-eligible 2740 terminals. Where applicable, the PAGDEL option must be specified (or used by default) for these terminals.

Message Format Service (MFS) can be used in an IMS Fast Path configuration.

The TERMINAL macro statement can be specified without operands if the terminal is a non-station control 2740-1, 2741, printer, punch, tape, or disk.

All non-VTAM data communication specifications must precede the VTAM macro set in your IMS system definition stage 1 input deck. You receive a stage 1 output warning message if the VTAM macro set is not the last physical set. If an MSC macro set is part of your system definition, it must precede the VTAM macro set, or your system definition does not complete. To add VTAM support, you must specify either an ON-LINE or ALL system definition on the IMSCTRL macro statement.

For VTAM terminals, the default values shown for operands of the TERMINAL macro statement are ignored if these operands are specified on the TYPE macro statement.

If system definition statements from previous releases of IMS are being used, TERMINAL macro definitions from the previous releases are valid for this release. The TERMINAL keywords in this section must be used for terminals newly supported with this release of IMS and for terminals that use new IMS terminal functions.

The TERMINAL macro statement can be used to define secondary logical units type 1, type 2, type 4, and type P, logical unit type 6.1, and NTO devices. These are designated on the TYPE macro as UNITYPE=SLUTYPE1, SLUTYPE2, SLUTYPEP, LUTYPE6, and NTO, respectively.

By specifying SLUTYPE1 on the TYPE macro, you can define an appropriate configuration of console, bulk printer, disk and card reader/punch. The 3767 and 3770 nonprogrammable terminals must be defined to IMS as SLUTYPE1.

SLUTYPE2 must also be specified for display devices attached to a 3274 or 3276 Control Unit that is operating in SNA mode.

**Related Reading:** For 3270/SLUTYPE2 dynamic terminals that are not defined by the TERMINAL macro, see MFS Device Characteristics Table Utility in *IMS Version 9: Utilities Reference: Database and Transaction Manager*.

By specifying SLUTYPEP on the TYPE macro, you can define 3600 terminals and 3790 or later programs using the Host Communication Facility. This support extends full IMS functional capabilities to user-written programs within the controller and includes the MFS Distributed Presentation Management (DPM) function. The support is in addition to current support for 3600 and 3790 user-written programs.

By specifying LUTYPE6 on the TYPE macro, you can define a logical unit type 6 node such as CICS, another IMS, or a user-written program. This node can communicate with IMS as an SNA primary or secondary half-session.

For logical units type 6, the log write-ahead option, as specified by the LTWA and NLTWA parameters on the TERMINAL macro statement, need no longer be specified, because log write-ahead occurs automatically. Therefore, although this parameter can be retained for compatibility with previous releases, if it is specified, it is ignored by IMS.

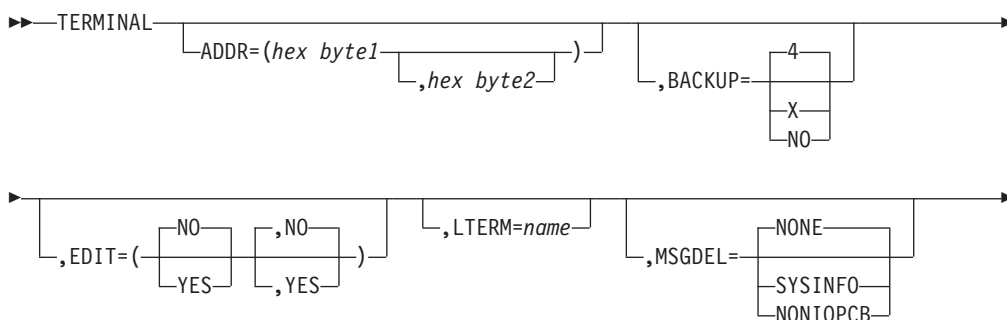
By specifying NTO on the TYPE macro, you can define NTO devices. NTO support provides a non-SNA start/stop terminal interface to VTAM for 3101, 2740-1, 2741, TTY, and TTY-compatible devices. The TERMINAL macro statement keyword PU= permits you to designate the type of terminal being defined to use NTO support.

**Related Reading:** For further information on SLU P and LU 6.1 (ISC), see the *IMS Version 9: Administration Guide: Transaction Manager*.

The TERMINAL macro statement can also be used to define terminals belonging to a Finance Communication System such as the 3600 Finance Communication System or the 4700 Finance Communication System. Finance Communication System components for devices specified as UNITYPE=3601 or as UNITYPE=FINANCE on the TYPE macro statement can be defined on the TERMINAL macro statement using either specific 3600 definitions or generic FINANCE terminal definitions. The code generated to support the terminal is identical regardless of whether the terminal is specified as UNITYPE=3601 or UNITYPE=FINANCE.

By specifying SLUTYPE2 on the TYPE macro, you can define 3277 terminals, Models 1 and 2, operating under control of the 3790/3270 Data Stream Compatibility. Under this support, the 3277 Model 2 can be defined as the IMS master terminal; the Model 1 is not supported as an IMS master terminal. This support is in addition to current support for the 3600 and 3790 user-written programs. The 3790 can be defined appropriately as SLUTYPE1, SLUTYPE2 or as a subset of FINANCE.

**All Non-VTAM Terminals**



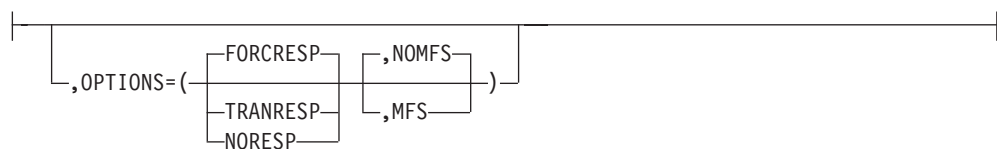




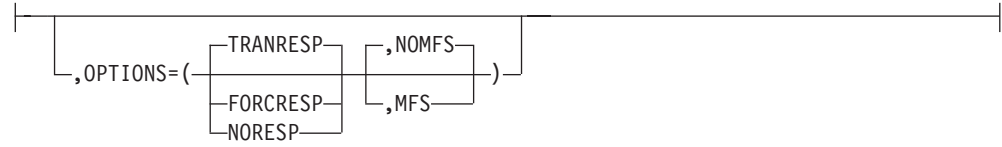
**Notes:**

- 1 FES not valid for LU 6 terminals
- 2 3270 Local Terminal
- 3 3270 Remote Terminal
- 4 Finance Work Station (If UNITYPE=FINANCE on the TYPE macro)
- 5 3600 Work Station (If UNITYPE=3601 on the TYPE macro)
- 6 SLU 1 Terminal
- 7 NTO Device
- 8 SLU 2 Terminal
- 9 SLU P Terminal
- 10 LU 6 Terminal

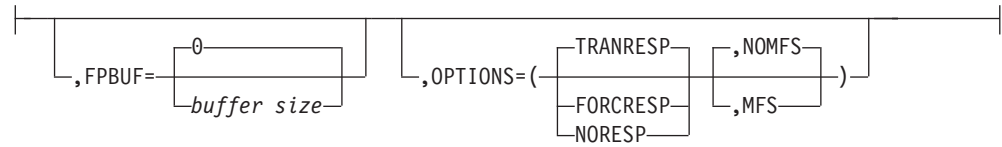
**A: (2740 Non-Station Control Terminal):**



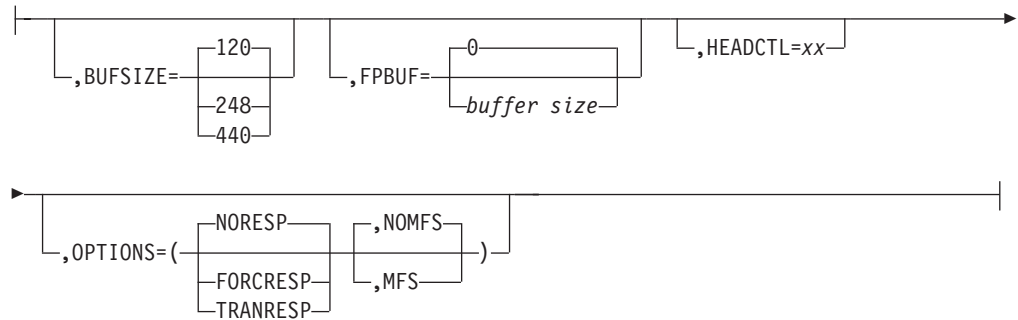
**B: (2740 Switched Terminal):**



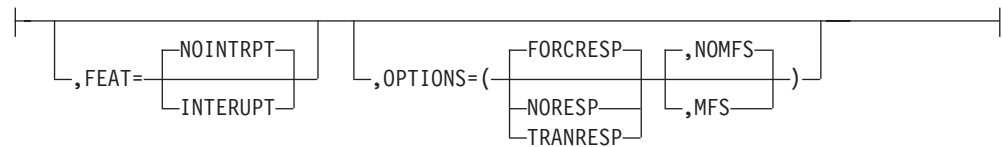
**C: (2740 Station Control Terminal):**



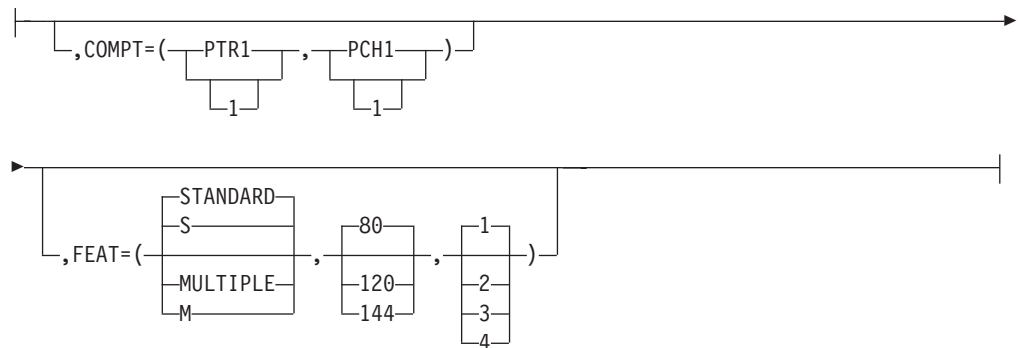
**D: (2740 Model 2 Terminal):**



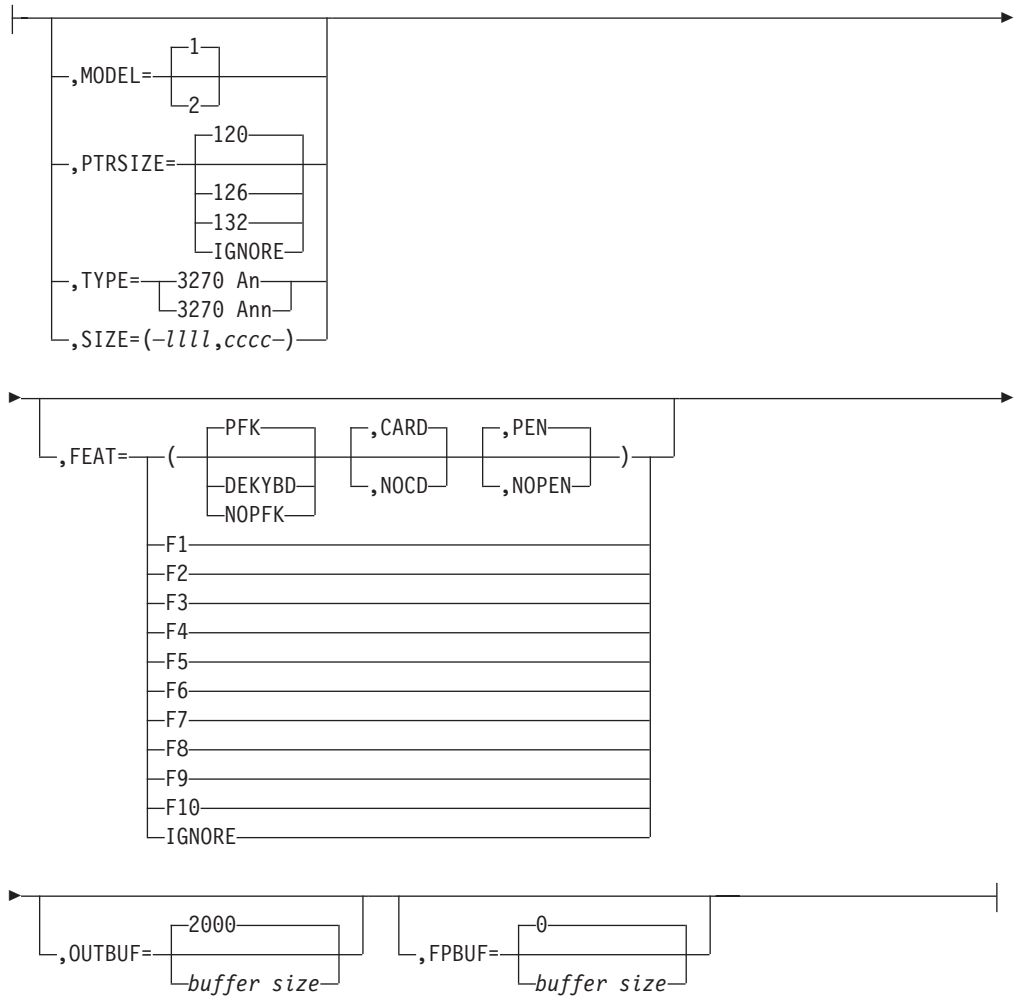
**E: (2741 Terminal):**



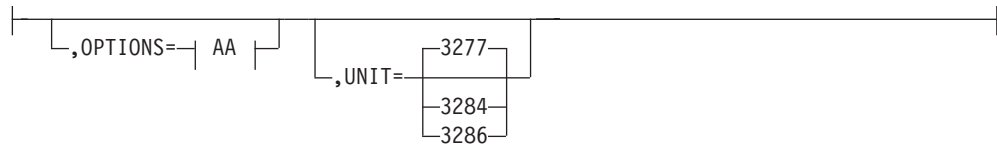
**F: (2780 Terminal):**



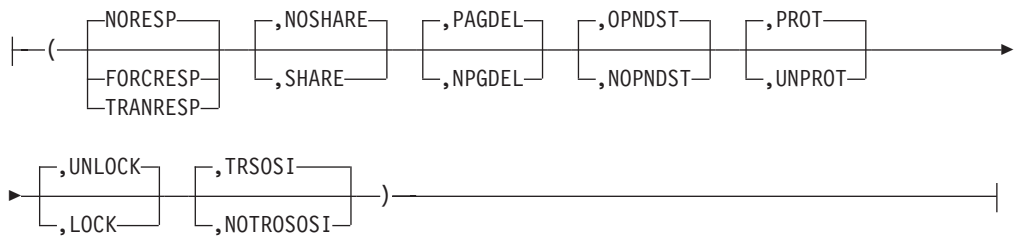
**G: (All 3270 Terminals):**



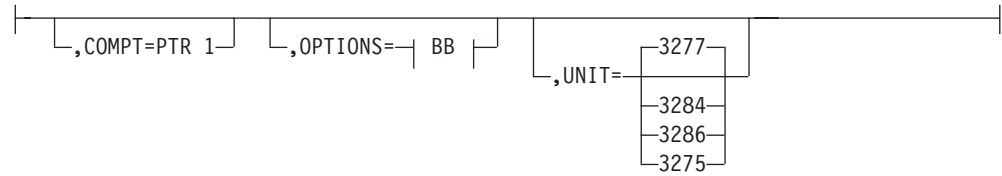
**H: (3270 Local Terminals):**



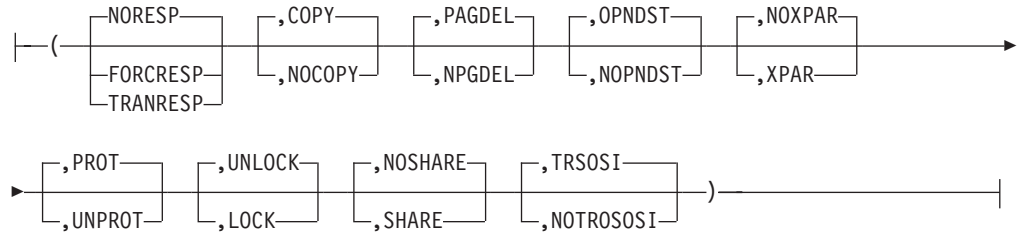
**AA:**



**I: (3270 Remote Terminal):**



**BB:**



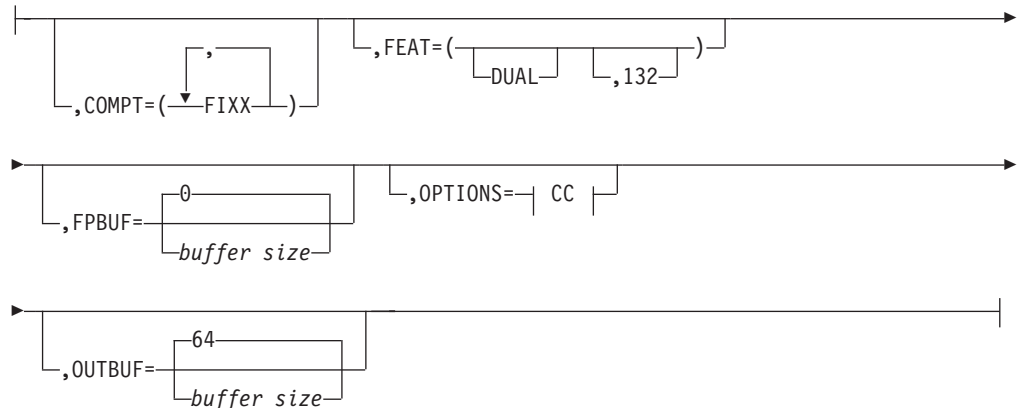
**J: (Spool Terminal):**



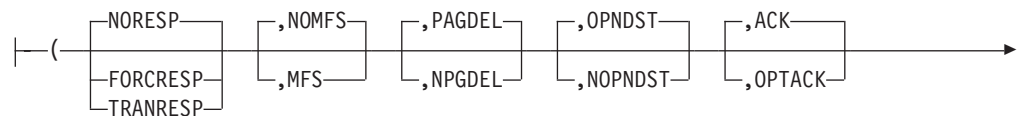
**K: (System/3 and System/7 Terminal):**

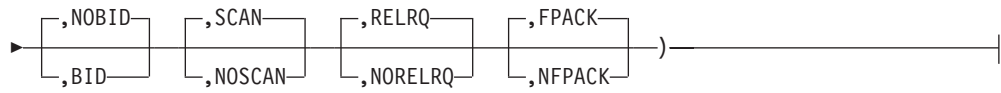


**L: Finance Work Station (UNITYPE=FINANCE on the TYPE macro):**

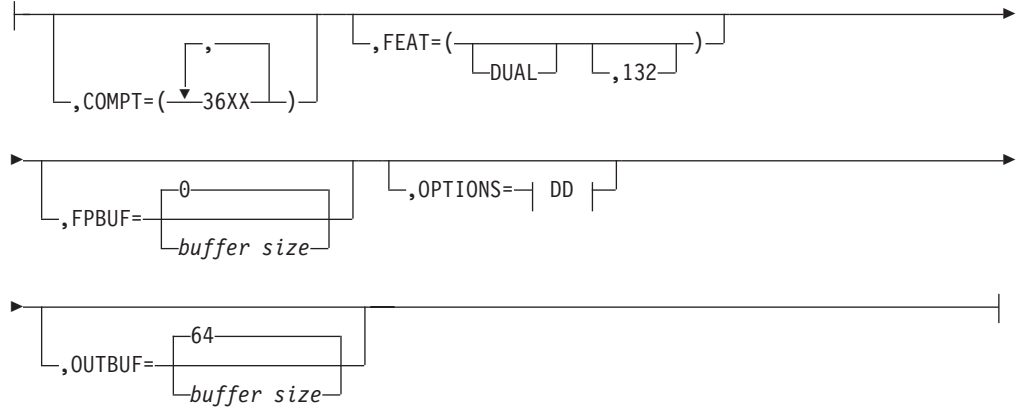


**CC:**

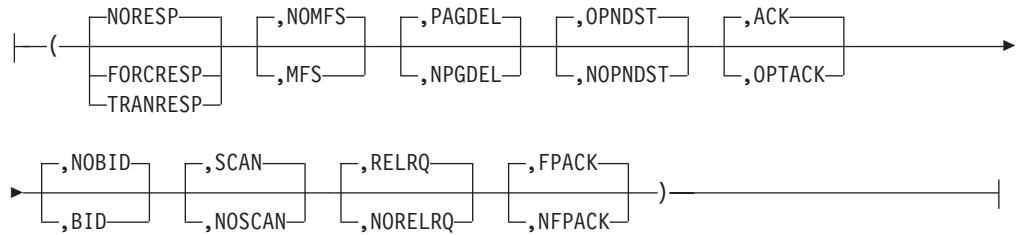




**M: 3600 Work Station (If UNITYPE=3601 on the TYPE macro):**

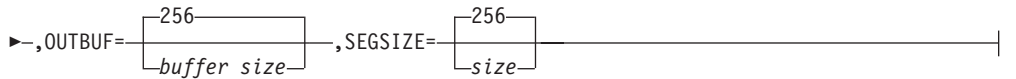
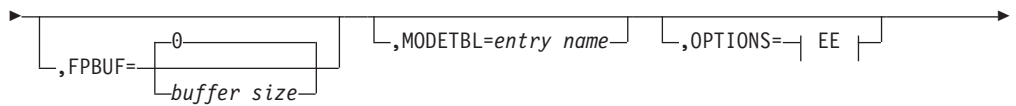
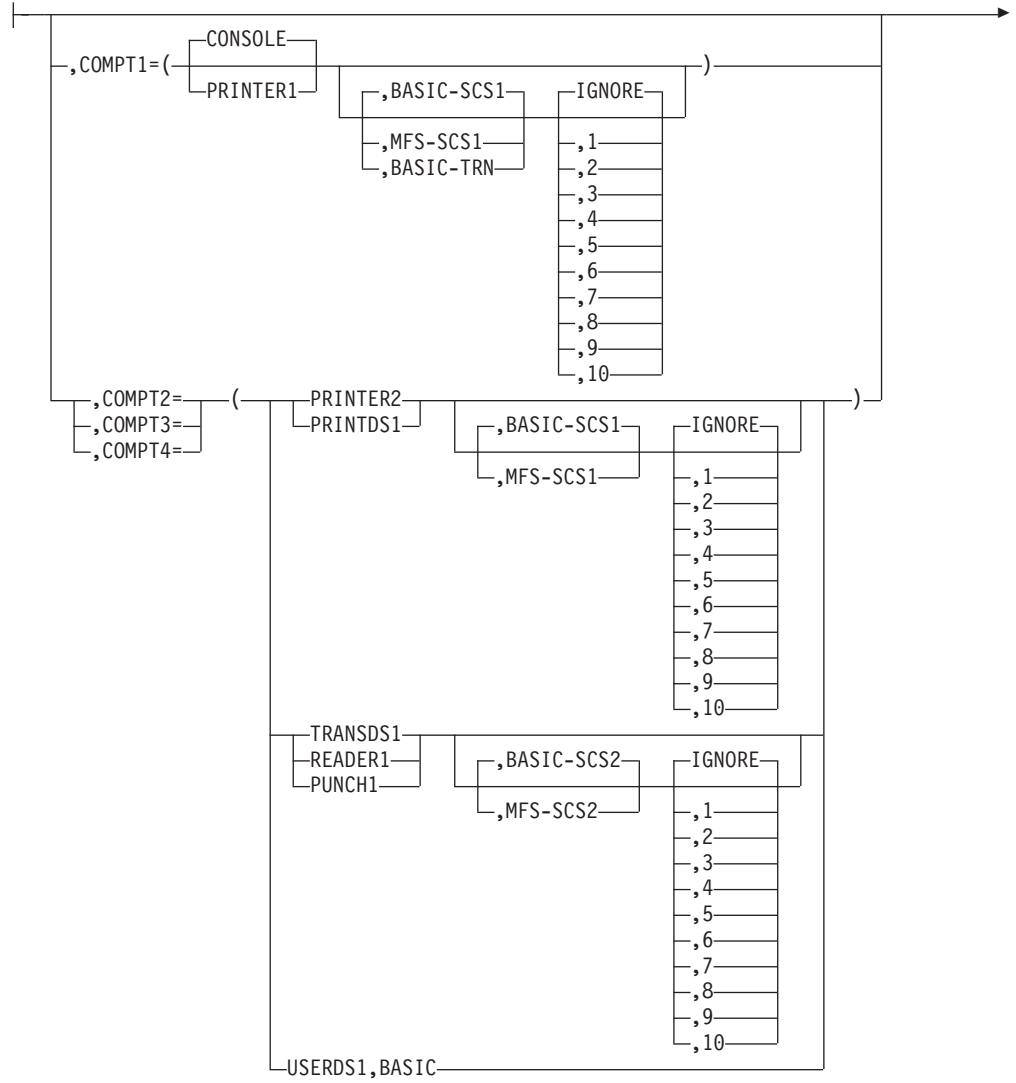


**DD:**

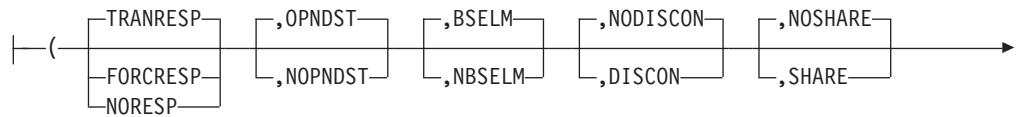


**N: (SLU 1 Terminal):**

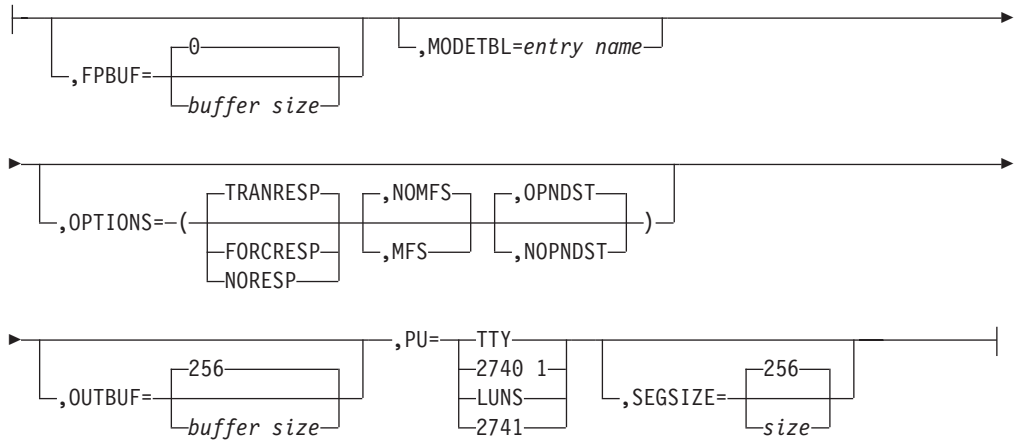




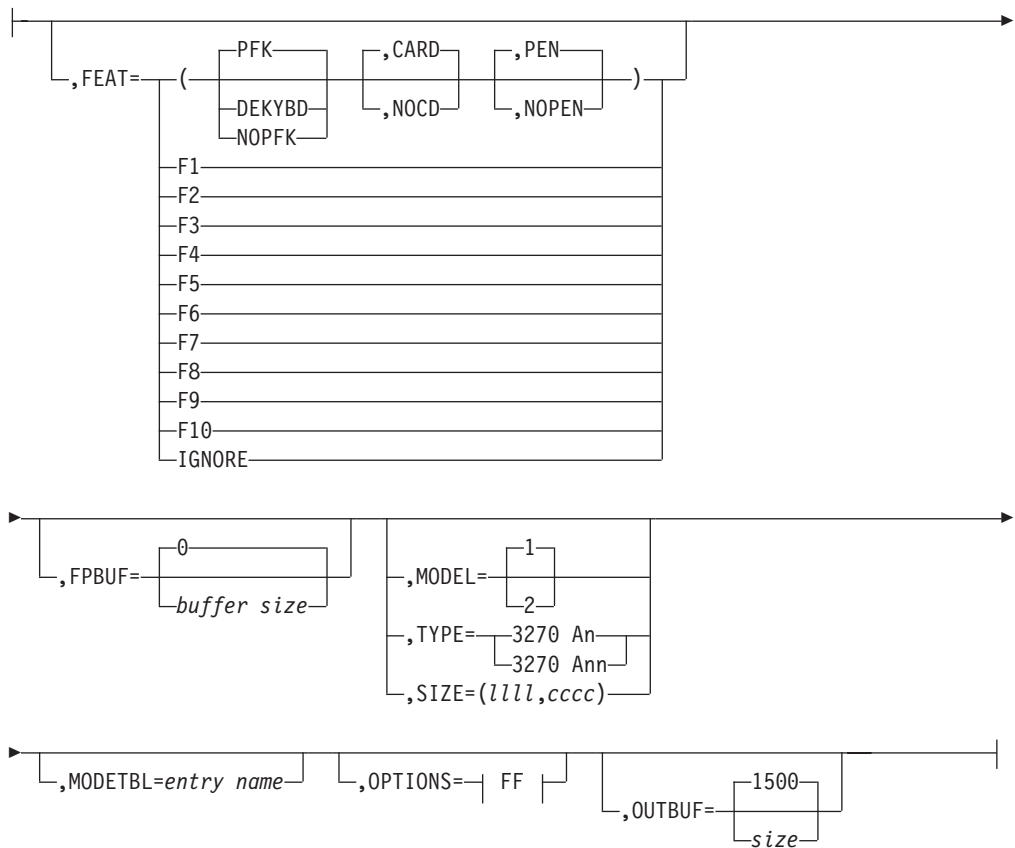
**EE:**



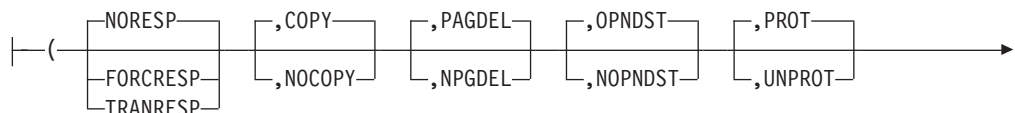
**O: (NTO Device):**



**P: (SLU 2 Terminal):**

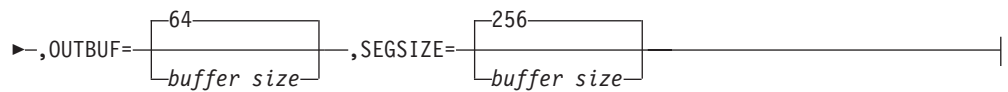
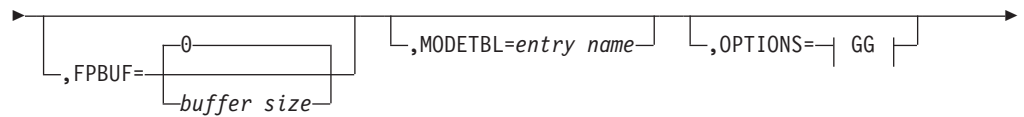
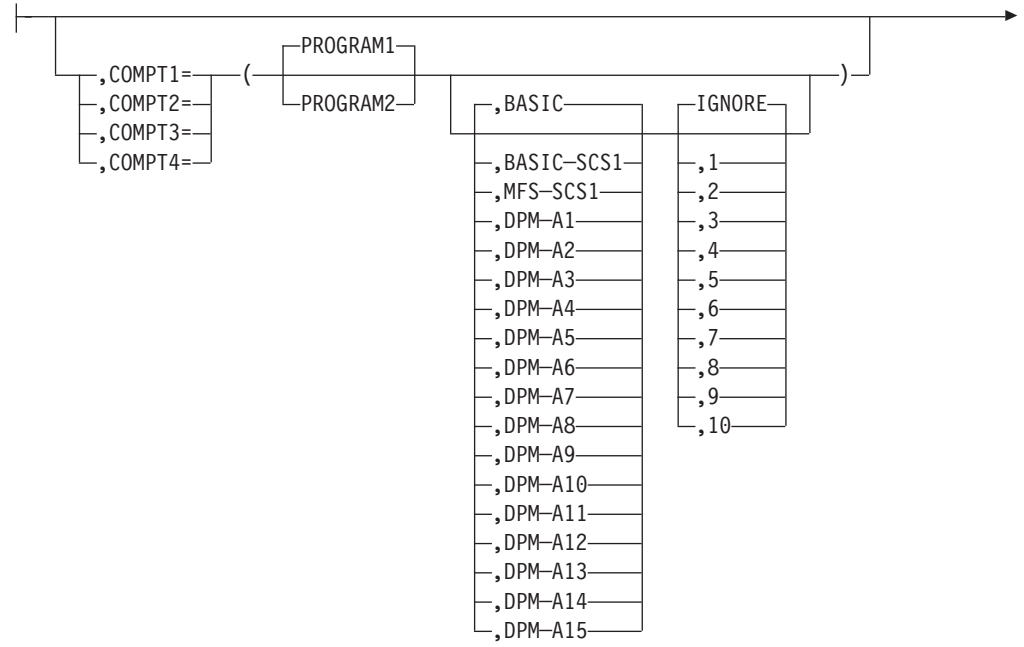


**FF:**

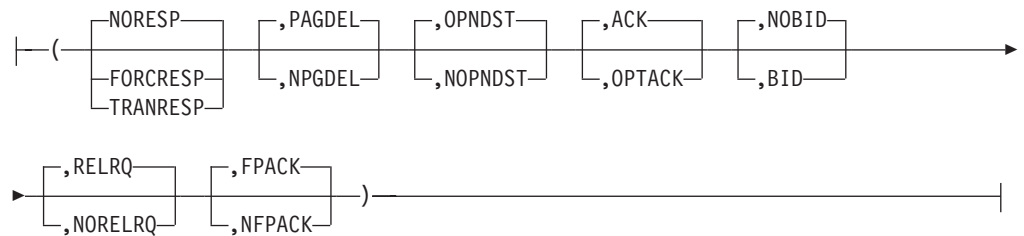




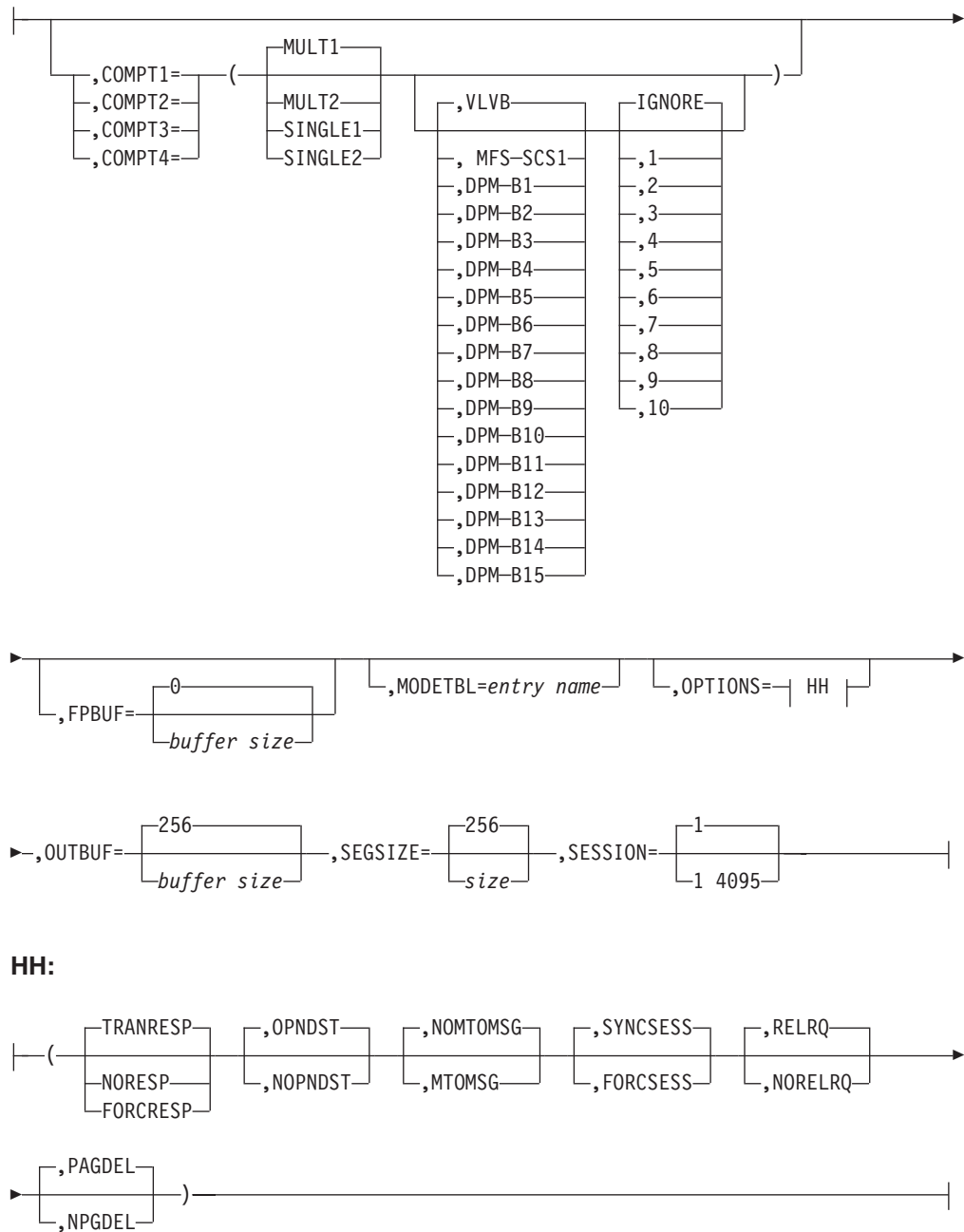
**Q: (SLU P Terminal):**



**GG:**



**R: (LU 6.1 Terminal):**



## Label Field

If any subsequent NAME macro statement refers to this TERMINAL macro statement in its OUTPUT keyword, a label must be specified for the TERMINAL macro statement.

## Keyword Parameters

To find which parameters apply to your IMS configuration refer to Table 15 on page 33.

### ADDR=

Specifies, when XRF is not used, the physical terminal addressing character in terminal code (hexadecimal representation) for all terminals. For example, physical terminal address "A" for a 2740 is coded ADDR=E2 (E2 is 2740 line

code for "A"). For specific terminal types, check the appropriate component description manuals for the types of addresses required.

If the terminal is a local 3270, the terminal address must be specified as four hexadecimal digits.

If the terminal is a System/3, System/7, or nonswitched 3275, the address must be specified as four hexadecimal digits. For a nonswitched 3275 or 3276 terminal, the first two digits specify the control unit polling address, while the last two specify the terminal address (example, ADDR=4040).

If the terminal is a 3277 or 3284/3286 attached to a 3271, the address must be specified as two hexadecimal digits, specifying the terminal address related to the particular display station or printer.

Independent of terminal type, all input terminal addresses for a line must be unique. This operand is not required for 2741, switched 3275, READER, PRINTER, PUNCH, TAPE, DISK, SPOOL, or 2740 Model 1 non-station control terminals. For all other terminal types not previously mentioned, the address must be specified as two hexadecimal digits.

When XRF is used, a second set of polling characters should be specified in the ADDR= keyword. This additional parameter is only used to define the second address for a remote BTAM-supported IMS master or secondary master terminal. And, when specified, only the master terminal and the secondary master terminal can be in the line group. RSR does not support BTAM terminals.

#### **BACKUP=**

Specifies (for an XRF complex) control of the automatic restart or session switching (VTAM) after takeover. Use only when HSB=YES is specified on the IMSCTRL macro.

X is a numeric integer from 1 to 7, inclusive, that specifies the priority for reestablishing the session. The default is 4 if either the keyword or the parameter is omitted. NO suppresses session recovery of the terminal at takeover. The priority specified is effective not only in switching and reestablishing a session during takeover, but also for all session activities performed under the terminal.

Although requests to VTAM are prioritized by IMS, the active requests can be completed in any order because of internal VTAM conflicts and pacing.

The second parameter is only used with VTAM terminals. It determines whether the backup system is to attempt to establish a backup session to the terminal when the active request establishes a session. Backup sessions are not valid for UNITYPEs 3270, LU6, and NTO. If this parameter is specified as YES for these UNITYPEs, it is ignored. If it is not specified, the default is NO. The default for this second parameter is YES when either the keyword or the parameter is omitted. Specifying NO for this parameter suppresses the establishment of a backup session.

#### **BUFSIZE=**

For 2740 Model 2 lines, specifies the size of the output buffer for this 2740 Model 2 terminal. The value specified here cannot exceed the value specified on the preceding LINE macro statement. The default is 120.

#### **COMPT=**

Specifies components of this terminal. This operand is valid only for 2780, nonswitched 3275, and 3600 terminals.

- For 2780 terminals:

Valid entries are PTR1 and PCH1, or 1 and 2 respectively. These entries can appear in any sequence. If this keyword specification is omitted, it is assumed that the user has a full complement of output components for the device.

- For nonswitched 3275 terminals:  
The only valid entry is PTR1. PTR1 specifies that a 3284 Model 3 printer is attached to this 3275 terminal. The video display is considered as component 1 and the 3284 Model 3 printer is considered as component 2.
- For terminals specified as UNITYPE=3601 or FINANCE on the TYPE macro:  
Describes the components that compose this workstation. A maximum of four terminal cans can be defined. Acceptable values for components are:

**36DS or FIDS**

3604 display or 4704 keyboard/display

**36DS3 or FIDS3**

3604 display, model 3

**36DS4 or FIDS4**

3604 display, model 4

**36DS7 or FIDS7**

3604 display, model 7

**36JP or FIJP**

3610/3612 journal printer or 4710 receipt validation printer

**36PB or FIPB**

3611/3612 passbook printer

**36FP or FIFP**

3618 line printer

**36MS or FIMS**

Magnetic strip encoder

**36CT or FICT**

3614 customer transaction facility (a component of FINANCE)

One 36DS, 36DS3, 36DS4, 36DS7, or 36CT (or equivalent components specified as FIDS, FIDS3, FIDS4, FIDS7, or FICT) can be specified per workstation. Multiple components of the same type (excluding 36DS, 36DS3, 36DS4, and 36DS7 or equivalent F<sub>ix</sub> components) can be specified. The order in which components are specified determines the value placed in the COMPT field of the Finance Communication System output message header.

**COMPT1=, COMPT2=, COMPT3=, COMPT4=**

- For SLU 1 Terminals:  
Defines the component types and processing associated with terminals defined as SLUTYPE1. A maximum of four components can be defined. For each component, three subparameters—media, edit processing, and features—can be defined.  
  
You can assign a number to a set of user-defined features (such as printer size, vertical forms control, horizontal tabs on a printer, and logical record length on a diskette) and tell MFS what these features are. IGNORE can be used to specify that all features (or any designated feature) are to be ignored.  
  
Component 1 must be either CONSOLE or PRINTER1. Components 2, 3, and 4 must be PRINTER2, PRINTDS1, TRANSDS1, READER1, PUNCH1, or USERDS1.

If no components are specified, the default is `COMPT1=(CONSOLE,BASIC-SCS1)`.

The first subparameter (media) of the second, third, or fourth components has no default.

If `BASIC-SCS1`, `BASIC-SCS2`, `BASIC`, or `BASIC-TRN` is specified as the second subparameter (editing), the third subparameter (feature) cannot be specified. If `MFS-SCS1` or `MFS-SCS2` editing is specified as the second subparameter, the default is `IGNORE` for the third subparameter.

The following values can be defined:

#### **CONSOLE (BASIC-SCS1 or MFS-SCS1)**

Identifies the presence of a console keyboard and a printer. Either this component or `PRINTER1` must be (and can only be) specified first in a sublist, because all error messages are routed to this destination. This component uses `SCS1` data streams. This can be the only component specified if the secondary logical unit type 1 is a 3767.

#### **CONSOLE (BASIC-TRN)**

Identifies a SLU 1 terminal whose input data does not conform to `SCS` (`SCS1` or `SCS2`) protocols. IMS processes the data without editing, translating, or deleting characters following the destination and password fields. No MFS editing is supported.

This form of transparency support differs from `SCS2` transparency support in that `BINDPSB1=BINTRNDS` in the SLU 1 bind image need not be set (and has no effect if set), and in that IMS does not scan the input data stream for transparency control characters (X'35') to identify transparent fields.

#### **PRINTDS1**

Identifies the presence of a print data set or the `SYS.INTR` data set for the SLU 1 programmable model. This component uses `SCS1` data streams.

#### **PRINTER1**

Identifies the presence of 3784, 3284, 3286, or similar printers. This component or console must be (and can only be) specified first in a sublist, because all error messages are routed to these destinations. This component uses `SCS1` data streams.

#### **PRINTER2**

Identifies the presence of a SLU 1 printer. This component uses `SCS1` data streams.

#### **PUNCH1**

Identifies the presence of a 3770 card output device. This component uses `SCS2` data streams. Card records do not span RU boundaries.

#### **READER1**

Identifies the presence of a 3770 card input device. This component uses `SCS2` data streams.

#### **TRANSDS1**

Identifies the presence of an input component consisting of the `SLUTYPE1` transmit data set. This component uses `SCS2` data streams. `TRANSDS1` can be specified only once as a component of a particular terminal.

#### **USERDS1**

Identifies the presence of the user's sequential data sets for output and

input. MFS support is not available for this component. The data stream on output is undefined; the inbound data stream is the same as the transmit data set data stream.

If multiple USERDS1 components are defined on the same TERMINAL macro statement, all user data set (UDS) input appears to be from the first UDS defined. Also, unless the CHNG call is used by the IMS Message Processing Program, all output is sent to the output component associated with the first UDS defined. (See the COMPT parameter of the NAME macro statement.)

If MFS-SCS1 or MFS-SCS2 data streams are specified, MFS-SCS1 or MFS-SCS2 formatting is used for messages processed by MFS.

- For SLU P nodes:

Defines the component types and processing (including distributed presentation management for MFS) associated with a node defined as SLUTYPEP, and indicates that the component is represented by a user-written program.

Up to four components can be specified; for each component, up to three subparameters can be specified. If the component is not specified, the default is COMPT1=(PROGRAM1,BASIC). Components 1, 2, 3, and 4 must be either PROGRAM1 or PROGRAM2. If component 2, 3, or 4 is present and takes all defaults (PROGRAM1,BASIC), at least one of the default subparameters must be specified so that the presence of the component can be recognized.

The first subparameter is the media type coded as PROGRAM1 or PROGRAM2.

If PROGRAM1 is specified, IMS does not assume component protection for this component and might send consecutive messages without waiting for intervening input requests. When using MFS, component protection can be accomplished for messages which use MFS by means of paging options in MFS control blocks defined as DEV TYPE=DPM-A01...DPM-A15 or DPM-A1...DPM-A15.

If PROGRAM2 is specified, IMS assumes component protection and does not send consecutive messages without intervening input requests.

The second subparameter specifies the type of editing to be provided by IMS for input to and output from user-written programs.

If BASIC is specified, no deblocking occurs on input, and MFS is not used on output or input.

If DPM-A01...DPM-A15 or DPM-A1...DPM-A15 is specified, no deblocking occurs on input, and the MFS DPM function can be used on output or input. To define DPM for this component, code DPM-An, where "n" is a decimal number from 1 to 15. When MFS is to be used, this operand must match an MFS DEV TYPE=DPM-An statement within the MFS definitions.

If BASIC-SCS1 is specified, SCS1 data streams are used. MFS is not available on output or input.

If MFS-SCS1 is specified, SCS1 data streams are used, and MFS is available.

The third subparameter specifies a user-defined feature code. The designated feature is used to select an MFS format—MFS-SCS1, DPM-A01...DPM-A15, or DPM-A1...DPM-A15—with matching feature specification. IGNORE can be used to specify that an MFS format (SCS1 or DPM-An) with the FEAT=IGNORE operand of the DEV statement is to be selected.



**Related Reading:** See *IMS Version 9: Utilities Reference: Database and Transaction Manager* for additional information on the MFS format.

The third subparameter cannot be specified if BASIC or BASIC-SCS1 is specified for the type of editing (second subparameter).

- For LU 6.1 nodes:

Defines the SNA send/receive and bracket protocols to be used during asynchronous IMS output to LU 6.1 nodes. The MFS distributed presentation management (DPM) facility can be defined as available selectively by component.

The first, second, and third subparameter are specified independently of one another.

The first subparameter (MULT1, MULT2, SINGLE1, SINGLE2) indicates the output protocol to be used and the number of messages to be sent on output. These subparameters are defined as follows:

- MULT1—Multiple messages or empty queue ends the bracket
- MULT2—Multiple messages or empty queue causes the flow to be returned to the other half-session
- SINGLE1—A single message or empty queue ends the bracket
- SINGLE2—A single message or empty queue causes the flow to be returned to the other half-session

The default is MULT1.

**Related Reading:** Guidance on specification of a component as SINGLE1, SINGLE2, MULT1, or MULT2 is found in the *IMS Version 9: Administration Guide: Transaction Manager*, and the characteristics of SINGLE1, SINGLE2, MULT1, MULT2 are described in *IMS Version 9: Customization Guide*.

The second subparameter (VLVB/DPM-Bn) indicates whether MFS is available for the component and whether the data blocking algorithm is to be used. VLVB indicates variable-length, variable-blocked format is to be used in place of MFS for both input and output. DPM-Bn indicates that MFS DPM is available for both input and output from that component on a message-by-message basis. The default is VLVB. You should not specify the DPM-B(xx) operand for the XRF/ISC link. If DPM-B(xx) is specified, a G732 warning message is issued, and the default value of VLVB is assumed.

The third subparameter, a number from 1-10, specifies a user-defined feature code. The designated feature is used to select an MFS format. IGNORE can be used to specify that an MFS format with the FEAT=IGNORE operand of the DEV statement is to be selected. The default is IGNORE. (If the second subparameter is specified as VLVB, the third subparameter cannot be specified.)

**Related Reading:** For more information, refer to *IMS Version 9: Utilities Reference: Database and Transaction Manager*.

#### **EDIT=**

The first parameter specifies whether (YES) or not (NO) the user-supplied physical terminal output edit routine specified in the TYPE or LINEGRP macro statement is to be used for this workstation. The default is NO.

The second parameter specifies whether (YES) or not (NO) the user-supplied physical terminal input edit routine specified in the TYPE or LINEGRP macro statement is to be used for this workstation. The default is NO.

#### **FEAT=**

(If the system definition statements from previous releases of IMS are used, TERMINAL macro statements from previous releases are valid for this release.)

- For 2741 terminals:  
With the parameter INTERRUPT or NOINTRPT (the default), specifies whether the 2741 terminal is equipped with the Receive Interrupt feature.
- For 2780 terminals:  
With the indicated parameters, specifies 2780 terminal features.  
The first subparameter, (MULTIPLE or M) or (STANDARD or S), specifies the MULTIPLE or STANDARD record transmission feature. The default is STANDARD.  
The second subparameter specifies the size of the print line. The default is 80.  
The third subparameter specifies the terminal model number. The default is 1.
- For nonswitched or local 3270 terminals:  
The first subparameter specifies whether a DEKYBD (Data Entry Keyboard) or PFK (Program Function Keys) is available on this 3275 or 3277 terminal. PFK and DEKYBD are not valid, and NOPFK (no Program Function Keys) is forced for 3284 and 3286 terminals. The default is PFK.  
The second subparameter specifies whether the Operator Identification Card Reader is available (CARD) or not (NOCD) on this 3275 or 3277 terminal. CARD specification is not valid, and NOCD is forced for 3284 and 3286 terminals. The default is CARD.  
The third subparameter specifies whether (PEN) or not (NOPEN) the Selector Pen is available on this 3275 or 3277 terminal. PEN specification is not valid and NOPEN is forced for 3284 and 3286 terminals. The default is PEN.
- For terminals defined as SLUTYPE2:  
The comments for nonswitched and local 3270 terminals apply equally to terminals defined as SLUTYPE2.
- For all 3270 Terminals:  
IGNORE can be specified if the presence or absence of the value does not affect terminal operation within IMS.  
If IGNORE is not specified, the default values assumed for any omitted subparameters are PFK, CARD, and PEN, respectively.  
The CONFIG macro is used to define these features for switched 3275s.
- For all 3270 terminals (with the exception of the printer component of the 3275) and secondary logical units type 2 (SLU 2):  
F1, F2...F10 specify terminal features that the user designates to correspond to each integer. The same number is used for any given feature during both system definition and MFS processing. This specified integer is used instead of, and not in addition to, other feature specifications.  
If you are using the same MFS format for the 3180 and 3290, Fn must have different values for the 3180 and 3290. These same values must be coded on the corresponding FEAT=Fn parameter of the MFS DEV statements.
- For 3275 printer components:  
FEAT=F1...F10 is not a valid specification for 3275 printer components (that is, devices specified as UNIT=3275 and COMPT=PTR1).
- For 3284 and 3286 units:  
F1...F10 specify terminal features that the user designates to correspond to each integer. The same number is used for any given feature during both

system definition and MFS processing. This specified integer is used instead of, and not in addition to, other feature specifications.

Specifications of FEAT=F1...F10 and PTRSIZE are mutually exclusive for the 3284 and 3286 devices. If PTRSIZE= is specified, FEAT= is invalid. If FEAT= is specified, no default PTRSIZE is assumed.

Also, FEAT=IGNORE is invalid for these devices. PTRSIZE=IGNORE is used instead.

The use of a particular specification for F1...F10 should be directly related to a particular printer line size and should be consistently used for printers having that line size and identical attributes throughout your system definition.

**Example:** FEAT=F5 might be used to define printers with a line size of 126 print positions and color, FEAT=F6 for printers with a line size of 126 print positions but without color, and FEAT=F7 for printers with a line size of 120 or 132.

- For UNITYPE=3601 or FINANCE and COMPT=36FP:

DUAL indicates the presence of the DUAL forms control feature, and 132 indicates the presence of the extended print line feature.

- For SPOOL Terminals:

If FEAT=AUTOSCH, the print utility is automatically scheduled when the spool lines are stopped, when a spool data set fills, or when IMS is initialized. If FEAT=NOAUTOSCH (the default), the MTO must schedule the print utility.

- For System/3 and System/7:

When FEAT=INPUT, the terminal is treated as an input-only terminal and cannot receive output.

#### **FPBUF=**

Specifies the Fast Path terminal buffer size for the following terminal types: 2740 station control, 2740 Model 2, 3270 (BTAM and VTAM), 3601, SLU 1, SLU 2, NTO, SLU P, and LU 6.1. FPBUF provides the system default for the EMH buffer if the EMHL parameter is not specified. FPBUF=0 means that the terminal is not eligible for Fast Path processing. FPBUF=0 is the default. This specification can also be used to override an FPBUF= default on the TYPE macro statement. If the terminal is eligible for Fast Path transaction processing, the minimum value that can be specified is 12. The maximum value is 30632. The value specified must not be greater than the logical record length of the long message queue data set specified in the RECLNG= parameter of the MSGQUEUE macro.

This keyword is optional and cannot be specified on ETO terminal descriptors. See the RECLNG parameter of the MSGQUEUE macro for further information.

#### **HEADCTL=**

Specifies that the header control feature for this 2740 Model 2 terminal is valid only for 2740 Model 2 lines, and specifies the number of characters reserved for header control. If either the BUFSIZE or HEADCTL operand is omitted, the default value is the value specified on the preceding LINE statement. An entered value for header control cannot exceed 28 and must be a multiple of 4.

#### **LTERM=**

Specifies a logical terminal name for this TERMINAL statement. This name is used to check input terminal security for this terminal. The NAME macro referred to must precede this TERMINAL statement. When this operand is present, the TERMINAL statement cannot be followed by a NAME statement.

This operand is required for READER terminals. For example, a previously defined TERMINAL statement defined a 3270 remote terminal and was followed by the name ABLE. In addition, assume that this TERMINAL statement refers to a READER terminal that is an input-only device. All input security for the READER terminal is checked against the security characteristics assigned to ABLE.

If a READER terminal requires a dummy terminal definition for the NAME macro, code the dummy definition so that it is consistent with BTAM use in the system. The UNITYPE parameter of the LINEGRP macro determines whether BTAM is to be included.

This operand **cannot** be specified for NTO devices and is not supported for ACF/VTAM terminals.

When a list of LTERMs is defined under the previous TERMINAL statement to which reference is made, only the first LTERM name is selected from that list. If an attempt is made to select other than the first, IMS ignores the name chosen and assigns the value of the first name listed.

**Example:**

```

TERMINAL ADDR=E2
  NAME LTERM1
  NAME LTERM2
  NAME LTERM3

TERMINAL ADDR=E4,LTERM=LTERM2
    
```

In this example, IMS treats the second TERMINAL statement as though it reads LTERM=LTERM1.

**MODEL=**

Specifies the terminal model number for terminals with screen sizes of 480 (Model 1) or 1920 (Model 2) characters. Terminals with screen sizes other than 480 or 1920 characters are defined using the TYPE and SIZE keyword parameters. The default is 1. The model number specified for a 3277 for which OPTIONS=COPY is specified must be less than or equal to the model number specified for any candidate printer. The MODEL and TYPE/SIZE keyword parameters are mutually exclusive.

**MODETBL=**

| Specifies the name of the VTAM logon mode table entry (logon mode name)  
 | that contains the SNA bind parameters to be used when a session is initiated  
 | by the MTO or through the /OPNDST command.

This function allows a system definition specification for referencing an entry other than the default entry in the user's VTAM logon mode table. Normally, the terminal operator specifies this mode table entry name when logging on to a terminal; this is not possible, however, from the IMS master terminal because IMS initiates the session.

With this function, if MODETBL= is not specified at system definition, no functional or operational change affects the user.

If MODETBL= is specified at system definition, the specified entry name is used. The MODETBL can may be overridden by:

- The LOGON APPLID entry by the remote terminal operator
- VARY ACT, LOGON= command by the network terminal operator
- /OPNDST, /RST, or /CHANGE command by the MTO

The MODETBL= parameter is required for the IMS master terminal if the VTAM default mode table is not configured specifically for the device to be used as the IMS master terminal.

**Related Reading:** For device dependent considerations, see *IMS Version 9: Administration Guide: Transaction Manager*.

With MODETBL=, you do not need to specify the mode table entry when logging on to terminals that always require specification of the same mode table entry name.

### **MSGDEL=**

Specifies which message types IMS should discard for this terminal, if it is not the master terminal. The default is NONE for non-VTAM terminals and SYSINFO for VTAM terminals. NONE is ignored for VTAM. The MSGDEL option applies to a switched terminal only while that terminal is connected to IMS; any message queued while an LTERM is not connected to the switched line is not deleted.

#### **NONIOPCB | NOTERM | SYSINFO | NONE**

NONIOPCB specifies that IMS should discard the following message types destined for this terminal:

- Message switches
- Messages inserted by an application program to an alternate PCB
- /BROADCAST messages
- DFS059 TERMINAL status messages

When a switched terminal defined as MSGDEL=NONIOPCB is not connected to IMS, any message normally discarded as a function of specifying NONIOPCB is queued for an LTERM that might become associated with the switched terminal at signon (/IAM) time.

NOTERM specifies that IMS should discard DFS059 TERMINAL status messages and DFS3650 SESSION status messages for this terminal. Do not specify NOTERM for 3601, SLU P, or LU 6.1 VTAM terminals. If NOTERM is specified, a warning message is issued and the specification default is SYSINFO.

SYSINFO specifies that IMS should discard DFS059 TERMINAL status messages for this terminal.

NONE specifies that IMS should not discard any message types for this terminal.

When MSGDEL is specified, the following additional restrictions apply to the reassignment of logical terminals:

- Assignment of a logical terminal to a physical terminal defined as MSGDEL=NONIOPCB can only take place if no messages are in the Q3 (logical terminal's system) or Q4 (other than system) queues, unless that logical terminal is currently assigned to a physical terminal also defined as MSGDEL=NONIOPCB.
- Assignment of a logical terminal to a physical terminal defined as MSGDEL=SYSINFO or MSGDEL= NOTERM can only take place if no messages are in the system queue of the logical terminal, unless that logical terminal is currently assigned to a physical terminal defined as MSGDEL=SYSINFO, MSGDEL=NOTERM, or MSGDEL=NONIOPCB.

An LU 6.1 session can be initiated only if the MSGDEL specifications on both the TERMINAL macro statement and the SUBPOOL macro statement match. An LU 6.1 session is not established, and an error message is issued if these specifications are not identical.

**NAME=**

Must be the VTAM node name specified during VTAM/NCP definition. To avoid a possible conflict in using Security Maintenance Utility (SMU) STERM statements, do not use the names LINE and ALL as a VTAM node name.

When using XRF and VTAM-supported master terminals, a second node name should be specified for the master terminals. The two node names specify nodes in the two IMS XRF systems. The first is used by the system with HSBID=1 and the second by the system with HSBID=2. If a secondary master is used, it also should have two specifications. For more information on the HSBID parameter, see Chapter 4, "Procedures," on page 191.

When using RSR and VTAM-supported master terminals, a third node name can be specified for the master terminals. The three node names specify nodes in the IMS RSR subsystems. The first is used by the subsystem with MTOID=1, the second by the subsystem with MTOID=2, and the third by the subsystem with MTOID=3. If a secondary master is used, it also should have at least two specifications (for RSR without XRF) or three specifications (for RSR with XRF at the active site). Note that the third node name can only be used for the tracking subsystem. For more information see the description of the MTOID parameter for "Member DFSSRSxx" on page 366.

The specification of *nodenamen* corresponds to the HSBID= specification at IMS startup. This means that HSBID=2 uses *nodename2* as the master or secondary master terminal, and uses PASSWD2 (from COMM macro). RSR tracking subsystems must select *nodenamen* and PASSWDn using the MTOID= parameter in the DFSSRSxx member. Tracking subsystems can override the APPLID for the subsystem by using the APPLID3= startup parameter.

When you define an ISC surveillance link between the XRF active and alternate systems, the two node names in the NAME keyword must match the two VTAM application names in the APPLID keyword of the COMM macro.

For an ISC node communicating with an XRF complex, the node name must be the VTAM USERVAR associated with the XRF complex. Defining additional nodes for the actual VTAM APPLIDs of the IMS systems which comprise the XRF complex will result in errors during session initiation.

The recommended use of the *nodenamen* keyword is:

- Use *nodename1* only for DB/TM or DCCTL systems that are non-XRF and non-RSR
- Use *nodename1* and *nodename2* for DB/TM or DCCTL systems using XRF
- Use *nodename1* and *nodename3* for DB/TM or DCCTL systems using RSR
- Use *nodename1*, *nodename2*, and *nodename3* for DB/TM or DCCTL systems using both XRF and RSR

The default for *nodename3* is the value of *nodename1*.

**OPTIONS=**

Specifies certain communication options associated with this terminal. The OPTIONS= subparameters are not position dependent.

**ACK | OPTACK**

ACK indicates that recoverable and update transactions must be

acknowledged with a definite response 1 (FME) or definite response 2 (RRN). OPTACK allows input messages containing only a begin bracket indicator to be acknowledged by an end bracket indicator on the next output message. A definite response must be requested for recoverable input if ACK is specified, and is optional if OPTACK is specified. The use of OPTACK also affects the sequence number recording for message resynchronization.

**Related Reading:** Additional information on ACK and OPTACK is found in *IMS Version 9: Administration Guide: Transaction Manager*.

OPTACK is forced for Fast Path-eligible terminals if the value specified for FPBUF is greater than 0. ACK can be specified if BID is not also used. ACK allows a 3600/LUP terminal to add Fast Path without initially recoding the remote terminal program.

#### **BSELM | NBSELM**

Specifies whether (BSELM) or not (NBSELM) the backspace character elimination is required for input from a SLU 1 terminal.

#### **COPY | NOCOPY**

Specifies whether (COPY) or not (NOCOPY) the copy function is requested for this terminal. Terminals defined with the copy function must have appropriate candidate printers.

**Related Reading:** For a detailed discussion of candidate printers for 3270 terminals, refer to the discussion of controlling output in *IMS Version 9: Administration Guide: Transaction Manager*.

This subparameter is supported for dynamic SLU 2 terminals, but not for dynamic non-SNA VTAM 3270s.

#### **FPACK | NFPACK**

Applies only to 3601/Finance and SLU P terminal types and specifies whether (FPACK) or not (NFPACK) Fast Path output message protocol is to be used, as defined in *IMS Version 9: Administration Guide: Transaction Manager*.

The Fast Path output message protocol for FPACK normally requires another input (transaction or SNA RTR command), following transmission of the output Fast Path reply message, before any asynchronous output message can be sent. The NFPACK option indicates that standard non-Fast Path protocols are to be used, and allows any asynchronous output message to be sent immediately following acknowledgment of the Fast Path output reply message.

#### **NOASR | ASR**

For the Session Outage Notification facility only; specifies automatic session restart (ASR) on a VTAM node. The default is NOASR. Specifying ASR or NOASR on the TERMINAL macro overrides ASR definitions on the TYPE macro for a defined node.

Use the /DIS command to display the current ASR option, and the /CHANGE command to change it.

The ASR option is not supported for nodes that can have XRF backup sessions.

For LUTYPE6 nodes, the ASR option works only if IMS is acting as the primary logical unit (PLU).

**NOBID | BID**

NOBID indicates that the VTAM BID command is never to be used. BID indicates that the VTAM BID command always precedes output messages that occur while between brackets, regardless of response mode option used.

Prior to the availability of this option, IMS systems forced the equivalent of NOBID if NORESP was specified, and forced the equivalent of BID if FORCRESP is specified.

**NODISCON | DISCON**

Determines whether IMS should automatically terminate a session for a terminal defined as SLUTYPE1 when operating in unattended mode. (ATTEND or UNATTEND mode is determined from the logon mode table entry specified to be used during session initiation.) DISCON specifies that IMS should automatically terminate the communication session with the terminal after receiving all immediately available input or sending all available output. NODISCON suppresses that automatic termination. The logon mode table is described in the MODETBL= keyword description.

**NOFES | FES**

Specifies whether (FES) or not (NOFES) this terminal is capable of front-end switching. The default is NOFES.

To define a terminal as capable of front-end switching, define the front end switch exit routine with the FESEXIT keyword of the COMM macro. See "COMM Macro" on page 70 for more information about the FESEXIT keyword.

If FES is specified, the FES exit will get control for each input from this terminal.

**NOMFS | MFS**

Specifies whether (MFS) or not (NOMFS) the message format service is to be provided for this terminal. The default is NOMFS.

**NOMTOMSG | MTOMSG**

Specifies whether (MTOMSG) or not (NOMTOMSG) normal session initiation and termination messages are to be sent to the IMS MTO. The default is NOMTOMSG.

**NOSHARE | SHARE**

Specifies whether (SHARE) or not (NOSHARE) this printer can be shared between VTAM subsystems. If SHARE is specified, then IMS can issue a SIMLOGON VTAM macro when either of the following occurs:

- Data is queued for this printer.
- The VTAM 3270 PF12 key invokes the internal IMS copy function that selects this terminal, and the terminal is not connected to IMS.

If SHARE is specified, the OPNDST option must also have been selected for this terminal.

If SHARE is specified, a MODETBL entry cannot be specified for cross-domain resources. An entry of ALL X'00' in the VTAM MODETBL indicates to ACF/VTAM that the default session parameters are to be used for this terminal and that the terminal can be defined in any VTAM domain.

If the default option, NOSHARE, is specified, and if IMS releases this terminal to another VTAM application, the IMS MTO must issue an /OPN command, a VTAM VARY command, or a USS logon in order for IMS to reacquire the terminal.



This option is applicable to SLU 1, 3284/86 local, and VTAM 3270 devices only. If UNITYPE=3270 is specified on the TYPE macro statement, UNIT must be a 3284 or 3286 device.

### **NOSIGNON | SIGNON**

Specifies (for VTAM and BTAM terminals only) signon verification security processing (SIGNON). The default is NOSIGNON.

The SIGNON and NOSIGNON definitions in this macro override those definitions on the TYPE macro.

Any signon option that is specified during system definition for a static terminal can also be specified in the SMU. The SMU specifications are applied during IMS restart.

Instead of specifying OPTIONS=SIGNON in more than one system definition macro, you can require all static terminals to sign on by specifying a single parameter, SIGNON=ALL, in the DFSDCxxx PROCLIB member.

**Related Reading:** For more information about the SIGNON keyword, see "Member DFSDCxxx" on page 338.

### **NOXPAR | XPAR**

Specifies whether (XPAR) or not (NOXPAR) transparent data transmission is to be used for this 3270 terminal. XPAR indicates that MFS formatting can be bypassed for this terminal. This option applies only to BTAM remote 3270 terminals.

### **OPNDST | NOPNDST**

Applies only to VTAM terminals and specifies whether (OPNDST) or not (NOPNDST) the /OPNDST command is valid for this node. The default for all terminal types except NTO is that /OPNDST is valid.

The default value for NTO devices is NOPNDST. Either NOPNDST, the default value, or OPNDST is valid for nonswitched NTO devices. When using switched NTO devices, NOPNDST must be specified or accepted by default, because these devices must be logged on by an operator. An error results if OPNDST is specified for switched NTO devices.

The NOPNDST option is used to prevent the /OPNDST command from being used to start a session between IMS and a VTAM terminal.

This option does not apply to a VTAM terminal specified in the NAME macro as MASTER (the IMS master terminal), because that terminal is automatically opened.

### **PAGDEL | NPGDEL**

Specifies whether (PAGDEL) or not (NPGDEL) automatic page deletion is to be specified for this terminal. This option applies to 3270 terminals, SLU 2 terminals, SLU P nodes, and to 3601 workstations using MFS. PAGDEL is the default option, but the LU 6.1 nodes default is NPGDEL.

When the last physical page of the last logical page of the message is sent, the message is dequeued no matter which parameter is specified.

PAGDEL should be specified for terminals used in an interactive or conversational manner. If input from the terminal or node is normally followed by a response to the terminal or node, PAGDEL should be specified.

NPGDEL is provided for terminals that produce many input messages to which no response is made. When NPGDEL is specified, input from a

paged message is responded to by IMS with the first page of that same message. Any application program response is queued.

#### **PROT | UNPROT**

Specifies whether (PROT) or not (UNPROT) the terminal display is protected after each page of a message. If PROT is specified, the protected status of the terminal depends upon the unprotected screen mode option on the DSCA keyword of the MFS DEV statement or the SCA keyword of the MFS MFLD statement. This keyword applies only to 3270 display terminals and secondary logical units type 2 (SLU 2).

#### **RELRQ | NORELRQ**

Specifies whether (RELRQ) or not (NORELRQ) IMS is to release this terminal upon request from other VTAM subsystems. If NORELRQ is specified, requests by other VTAM subsystems for this terminal are not honored. The MTO must intervene to reassign this terminal to another VTAM subsystem. If RELRQ, the default, is specified, then IMS releases this terminal to other VTAM subsystems upon request.

This option is applicable to devices specified on the TYPE macro statement as UNITYPE=3601, FINANCE, SLUTYPE1, SLUTYPEP, and LUTYPE6, in addition to 3284 and 3286 printers.

NORELRQ is forced for 3277, 3275, SLU2, and NTO terminals.

#### **SCAN | NOSCAN**

Specifies whether (SCAN) or not (NOSCAN) IMS should edit 3601 device-control sequences (such as SELECT, POSITION, and WRITE TRANSPARENT) that would otherwise be split across two transmissions when a non-MFS formatted output segment must be split into multiple transmissions.

If an MFS-formatted output segment must be split across two transmissions, IMS edits 3601 device-control sequences (SELECT and POSITION) that would otherwise be split across two transmissions even if the NOSCAN option is specified.

**Related Reading:** See “Financial Communication System Programmer’s Guide” in *IMS Version 9: Administration Guide: Transaction Manager* for additional information.

#### **SYNCSESS | FORCSESS**

Specifies when session initiation is to be completed.

- SYNCSESS specifies that session initiation is to be completed only when session restart modes (message sequence numbers) agree.
- FORCSESS specifies that session initiation is to be completed regardless of the agreement of session restart modes.

The default is SYNCSESS. only when session restart modes (message sequence numbers) agree (SYNCSESS) or whether initiation should always be completed regardless of the agreement of session restart modes (FORCSESS). The default is SYNCSESS.

#### **TRANRESP | NORESP | FORCRESP**

TRANRESP specifies that the transaction code definition should be referred to determine whether response mode should be used. All Fast Path-eligible terminals operate in response mode—either TRANRESP or FORCRESP.

- TRANRESP is the default value for 2740-1, 3767, 3770 switched and station-control terminals, and for terminals defined as SLUTYPE1, NTO, or LUTYPE6.

- NORESP specifies that response mode is not to be allowed for this terminal. NORESP is the default for 2740-2, 3270, and 3601 terminals, and terminals defined as SLUTYPE2 and SLUTYPEP. NORESP is not compatible with Fast Path.
- FORCRESP specifies that response mode is to be forced for all transactions entered from this terminal. FORCRESP is the default for 2740 non-station control and 2741 terminals.

For BTAM and VTAM terminals, the combined specification of FORCRESP and NPGDEL is not recommended. Also, if MSGTYPE=RESPONSE is specified on the TRANSACT macro statement for these terminals, the combined specification of TRANRESP and NPGDEL is not recommended.

The keyboard of a display terminal defined with no automatic page deletion (NPGDEL) can become locked during execution if operating in terminal response mode. NPGDEL specifies that the current output message is not to be dequeued at the time of input. However, with terminal response mode, input is inhibited until the current output message is dequeued and the terminal response mode is reset. Thus, a terminal receiving multiple pages of output and defined with the combined specification of FORCRESP and NPGDEL might require master terminal intervention to reset the terminal response mode.

#### **TRSOSI | NOTRSOSI**

Specifies whether IMS translates the shift-out/shift-in (SO/SI) characters for MFS editing. The SO/SI characters indicate which characters are used for EGCS terminals. The default of TRSOSI should be used for non-EGCS terminals. This keyword only applies to 3270 terminals and type 2 secondary logical units (SLU2).

#### **UNLOCK | LOCK**

Specifies whether IMS is to unlock the terminal keyboard and reset the modified data tags after an MFS bypass.

- UNLOCK specifies that MODNAME=DFS.EDTN unlocks the terminal keyboard and resets the modified data tags
- LOCK specifies that an application program assumes responsibility for unlocking the terminal keyboard and resetting the modified data tags.

This keyword applies only to 3270 terminals and type 2 secondary logical units (SLU2).

#### **OUTBUF=**

Specifies the size of the IMS output buffer to be used for the workstations that are identified in this parameter description. The OUTBUF value indicates the maximum size of IMS message segments that are sent from IMS to the workstation in a single transmission.

For BTAM remote 3270 devices that are specified with data transparency (OPTIONS=XPAR), IMS will send a single transmission the size of OUTBUF. If the message exceeds the OUTBUF value, IMS will segment the message into 4K segments and send each segment individually, resulting in multiple transmissions for a single message.

For VTAM SNA devices, a single transmission represents a transmission from IMS to VTAM. For these devices, the output message is blocked (chained) based on the OUTBUF value and might result in multiple transmissions for a single message.

| For VTAM non-SNA devices, BTAM local 3270 devices, and remote 3270  
| devices that are specified without data transparency (OPTIONS=NOXPAR), IMS  
| will send the message in a single transmission. No blocking or chaining is  
| performed.

If MFS bypass is used, the OUTBUF value must be large enough to contain the entire output message. If the message that is inserted by the IMS application exceeds the OUTBUF value, the message is rejected by IMS. If MFS is used to edit the output message, the OUTBUF value is ignored and the message is sent regardless of its length. When sending to a printer, the OUTBUF value must be equal to or greater than the segment size, but less than 4096 bytes.

IMS also uses the OUTBUF value to set the transmission services usage field (byte 11) in the bind parameter. If this value is specified incorrectly, it might cause the device to reject the bind parameter. The value specified, plus the size of any routing information later appended by VTAM, must be less than the maximum message size specified for VTAM and its related NCPs. The output buffer size is calculated as follows for:

**3270:** Record size. The range of values that can be specified is 256 through 30720 bytes. The default is 2000 bytes. The actual hardware buffer capacity should be considered when creating the OUTBUF calculation. The hardware buffer maximum might be less than the default of 2000 bytes. For a device that is not a SNA 3270, OUTBUF is used only if you are using MFS bypass. Otherwise, OUTBUF is ignored.

**Related Reading:** For information on devices attached to a local 3274 control unit, see *3274 Control Unit Description and Programmer's Guide*.

**3601 or FINANCE:** Record size plus header. (Header is 3 bytes if MFS is not used or 29 bytes if MFS is used.) The range of values that can be specified is 64 to 30720 bytes. The default is 64 bytes.

**SLU 1:** Record size plus header. (The header is 25 bytes.) The range of values that can be specified is 128 to 30720 bytes. The default is 256 bytes.

**NTO:** The range of values that can be specified is 256 to 30720 bytes. The default is 256 bytes.

Output message segments are not blocked. Each segment is sent as an RU. Segments greater than the RU size are spanned, but an RU never contains more than one segment.

**SLU 2:** The range of values that can be specified is 256 to 30720 bytes. The default is 1500 bytes. A 3790 defined as SLUTYPE2 has a maximum value of 1536.

**SLU P:** Record size plus header. (The header is 5 bytes if MFS is not used and up to 42 bytes if MFS is used.) The range of values that can be specified is 64 to 30720 bytes. The default is 64 bytes.

**LU 6.1:** Record size plus header. The range of values that can be specified is 256 to 30720 bytes. The default is 256 bytes. The header size without MFS is variable from 8 to 45 bytes; with MFS, the header size is variable from 8 to 62 bytes.

**Related Reading:** For more information on LU 6.1 record size, see *IMS Version 9: Administration Guide: Transaction Manager*.

For all logical units that require a bind from IMS, the OUTBUF buffer size must be a decimal value that can be expressed by the algorithm  $X \text{ times } 2 \text{ to the power of } Y$ .  $X$  must be a value from 8 through 15 and  $Y$  must be a value from 3

through 11. This is a VTAM restriction. Thus, for example, the value 144 (representing  $9 \times 2^4$ ) and the value 28672 (representing  $14 \times 2^{11}$ ) are acceptable values. Details of this VTAM restriction are described in *ACF/VTAM Macro Language Reference*, with information on requesting unit sizes.

Valid VTAM buffer sizes are restricted to one of the following decimal values:

112	448	1792	7168	28672
120	480	1920	7680	30720
128	512	2048	8192	
144	576	2304	9216	
160	640	2560	10240	
176	704	2816	11264	
192	768	3072	12288	
208	832	3328	13312	
224	896	3584	14336	
240	960	3840	15360	
256	1024	4096	16384	
288	1152	4608	18432	
320	1280	5120	20480	
352	1408	5632	22528	
384	1536	6144	24576	
416	1664	6656	26624	

IMS converts the value given here to the format required for the bind parameter fields and places the value into the appropriate field in the bind parameter list. If the maximum RU value specified cannot be converted exactly into the bind format, the value is rounded down to the next-lower value bind format for inbound RUs and rounded up to the next-higher value bind format for outbound RUs.

**Note:** For IMS to CICS ISC connections, IMS checks the ISC link buffer definitions during the bind to avoid potential storage corruption that could occur by receiving a message that is larger than the available buffer. If a mismatch is detected, IMS issues message DFS2066I.

To avoid this conflict, ensure that:

1. If CICS is the primary half session, the CICS RECEIVESIZE is greater than or equal to the IMS OUTBUF size.
2. If IMS is the primary half session, the CICS SENDSIZE is greater than or equal to the OUTBUF size.
3. VTAM buffer sizes are a valid decimal value (expressed by the algorithm  $x$  times  $2$  to the power of  $y$ , with  $x = 8$  through  $15$ , and  $y = 3$  through  $11$ ). If valid VTAM buffer sizes are not specified, unwanted rounding of those values might occur, and a mismatch in bind RU sizes might result.

#### **PTRSIZE=**

Specifies the number of print positions of the 3284 or 3286 printer. The default is 120.

PTRSIZE=IGNORE causes MFS to always use the device output format (DOF) with FEAT=IGNORE in its DEV statement when editing output for this device. The existing parameters and control block values, including default parameters, are unchanged.

#### **PU=**

Specifies the type of physical device that is connected by means of the Network Terminal Option (NTO) licensed program. PU= is a required keyword for NTO devices; an error message results if it is omitted.

Each keyword represents a device type.

- TTY—TTY devices, such as a 33/35, or TTY compatible devices, such as a 3101 running in TTY mode.
- LUNS—These devices are non-SNA logical units, such as a 3101 not running in TTY mode.

**SEGSIZE=**

Specifies the maximum size of IMS message segments that are sent from the Adapter to IMS. Acceptable values are from 256 to 32000 bytes. The default is 256.

If an APPC logical record exceeds the SEGSIZE value, IMS will break the record into multiple IMS segments. The IMS application must issue get Next (GN) DL/I calls to receive the 2nd through nth segments.

Calculate the segment size as follows:

- SLU 1, SLU P, and NTO devices:  
Size is the maximum input segment size that can result from blocking or deblocking one or more SCS1 input records.
- LU 6.1:  
Size equals the size of the largest input record that can result from blocking or deblocking one or more VLVB or chain records.

**SESSION=**

For LU 6.1 sessions, specifies the maximum number of parallel sessions for the logical unit defined in the TERMINAL macro statement. From 1 to 4095 parallel sessions can be established. The default is 1.

When defining the XRF ISC SURVEILLANCE link, set the number of parallel sessions equal to one, or eliminate the SESSION= parameter.

**SIZE=**

Specifies the physical screen size of this 3270 display terminal having the device type symbolic name in the TYPE= operand. MFS uses the specified screen size to verify format specification. The first and second positional parameters of the operand relate to screen lines and columns, respectively. Both numbers must be specified as decimal numbers having the following range:

1111 = 1 through 16384  
 cccc = 2 through 16384  
 product of 1111 x cccc = 80 through 16384

where *1111* is the number of lines and *cccc* the number of columns available on the screen.

For 3270 devices defined as SLUTYPE2, the maximum number of lines and the maximum number of columns that can be defined for each is 255. Any symbolic name can be associated with any size screen. However, after this association is defined, the same device type symbolic name must be used. That is, it must duplicate a previously specified TYPE= specification of the LINEGRP, TYPE, or TERMINAL statement.

Specification of the physical screen size is required for the first (or only) definition of the specific device type symbolic name and can be omitted for subsequent identical device type symbolic name specifications. If omitted, the SIZE keyword operand originally specified with the device type symbolic name is used when the symbolic name is repeated. If the SIZE keyword is specified, the screen size defined must be the same as that defined for the original device

type symbolic name specification. You should establish a standard for relating device type symbolic names to the physical screen sizes.

The standards listed in Table 23 are recommended:

Table 23. Standards to Relate Device Names to Screen Sizes.

User-Defined Symbolic Name	Screen Size
3270-A1 or 3270-A01	12x80
3270-A2 or 3270-A02	24x80
3270-A3 or 3270-A03	32x80
3270-A4 or 3270-A04	43x80
3270-A5 or 3270-A05	12x40
3270-A6 or 3270-A06	6x40
3270-A7 or 3270-A07	27x132
3270-A8 or 3270-A08	62x160

For the 3270 display devices defined with the device type symbolic name, the system definition produces an MFS Device Characteristics Table (DFSUDT0x) where x is the suffix specified in the SUFFIX keyword of the IMSGEN macro statement. You can also use the MFS Device Characteristics Table (DCT) utility (DFSUTB00) to build and update this table. This table is stored in IMS.SDFSRESL and is used by the MFS Language Utility and the IMS online to extract the information associated with the specific device type symbolic name.

**Important:** The MFS DCT must include each combination of device type symbolic name, features, and screen size for all 3270 and SLU-2 terminals that are to be connected to IMS. Because ETO terminals are not defined in the IMS system definition, if DCT entries for ETO 3270 and SLU-2 terminals are not already in the DCT as the result of system definition statements for static terminals, use the MFS DCT utility to add the entries.

If the MFS Language utility (DFSUPAA0) cannot find the correct entry in the MFS DCT, based upon device type symbolic name (3270-An) and features, then the MFS Language utility produces error message DFS1862 or DFS1863. When a user logs on to an ETO 3270 or SLU-2 terminal that is connected to IMS, and VTAM provides the screen size to IMS, IMS uses the screen size and the features that are specified on the logon descriptor to search the MFS DCT. IMS searches the MFS DCT to determine the device type symbolic name. If the search is unsuccessful, the logon attempt is rejected with messages DFS3646 and DFS3672.

#### **Related Reading:**

- Refer to *IMS Version 9: Utilities Reference: Database and Transaction Manager* for additional information on the MFS Language utility and the MFS DCT utility.
- Refer to *IMS Version 9: Application Programming: Transaction Manager* for more information on the Device Characteristics Table

MNOTE statements in the system definition stage 1 output listing denote each device type symbolic name specified for 3270 and SLU type 2 devices with their corresponding screen size values.

**Example:**

**Display Device Symbolic Names and Screen Sizes**

Symbolic Name	Screen size	
	Lines	Columns
3270-A01	12	80
3270-A02	24	80
3270-A03	32	80
3270-A04	43	80

After a device type symbolic name (for example, 3270-A03) is associated with a screen size (for example, 32x80), the size need not be specified and must not be changed for any TYPE or TERMINAL macro statement for which the same device type symbolic name is used. During a subsequent system definition, the MFS Device Characteristics table and terminal control blocks can change if terminal specifications are altered. You should examine your MFS formats to determine whether they are affected by the changes and recompile them if necessary.

**TYPE=**

Specifies that an alphanumeric symbolic name for a device type is to be associated with this IMS-supported BTAM or VTAM 3270 physical display terminal. The symbolic name can be of the form 3270-*Ann*, where *nn* is a value from 01 through 15. The leading zero can be omitted. The physical screen size of the 3270 display device associated with the device type symbolic name is specified in the SIZE= operand. Any 3270 display device can be defined with a device type symbolic name. 3270 display devices having 480 or 1920-character screen sizes can be defined using the MODEL keyword parameter rather than the TYPE and SIZE keyword parameters. However, 3270 display devices with other than 480 or 1920 characters must be defined using the device type symbolic name and, therefore, TYPE and SIZE. The TYPE/SIZE and MODEL keyword parameters are mutually exclusive.

**UNIT=**

Specifies the terminal for the previously defined line group. For 3270 line groups, UNIT= is used to define this terminal as a 3284 or 3286 printer or 3277 display unit. 3277 display units cannot be defined in the same local line group as 3284 or 3286 printers. This operand is also used to define a 3275 terminal that is part of a 3270 remote line group.

**TRANSACT Macro**

The TRANSACT macro statement is used one or more times with each APPLCTN macro statement to identify transactions as IMS exclusive, IMS Fast Path potential, or IMS Fast Path exclusive. It specifies the transaction codes that cause the application program named in the preceding APPLCTN macro to be scheduled for execution in an IMS message processing region. It also provides the IMS control program with information that influences the application program scheduling algorithm. It can define a message editing routine.

If this macro statement is preceded by an APPLCTN macro defining it as remote (specifies SYSID), this transaction is generated as a remote transaction.

An IMS Fast Path–exclusive transaction is identified by a TRANSACT macro statement following an APPLCTN statement that specifies both PGMTYPE=TP and FPATH=YES.



TRANSACT macro statements that identify IMS Fast Path–exclusive transactions generate routing code entries. These entries have the same value as the corresponding transaction codes. They are used by the IMS Fast Path Expedited Message Handler routines to locate the Fast Path application program named in the preceding APPLCTN macro statement that processes the input message.

The SMU cannot define a Fast Path–exclusive transaction as able to issue commands.

An IMS Fast Path potential transaction is identified by a TRANSACT macro statement that specifies FPATH=YES following an APPLCTN statement that specifies PGMTYPE=TP and FPATH=NO.

An application defined as FPATH=NO cannot be run as a Fast Path region (IFP), even though it contains Fast Path potential transactions. Another application, defined with FPATH=YES, must exist and one of the following:

- A TRANSACT macro for the transaction
- A routing code defined by the RTCODE macro for the transaction

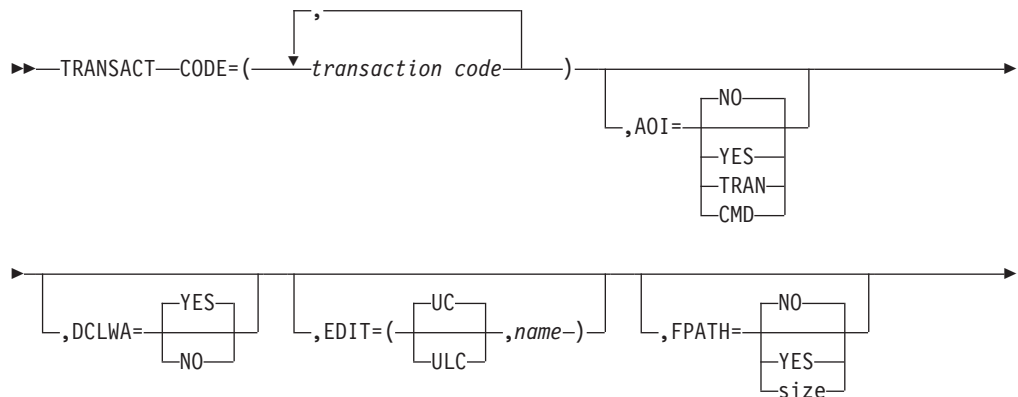
**Related Reading:** See the description of the Fast Path Routine for the Input Edit/Routing exit routine in the *IMS Version 9: Customization Guide*.

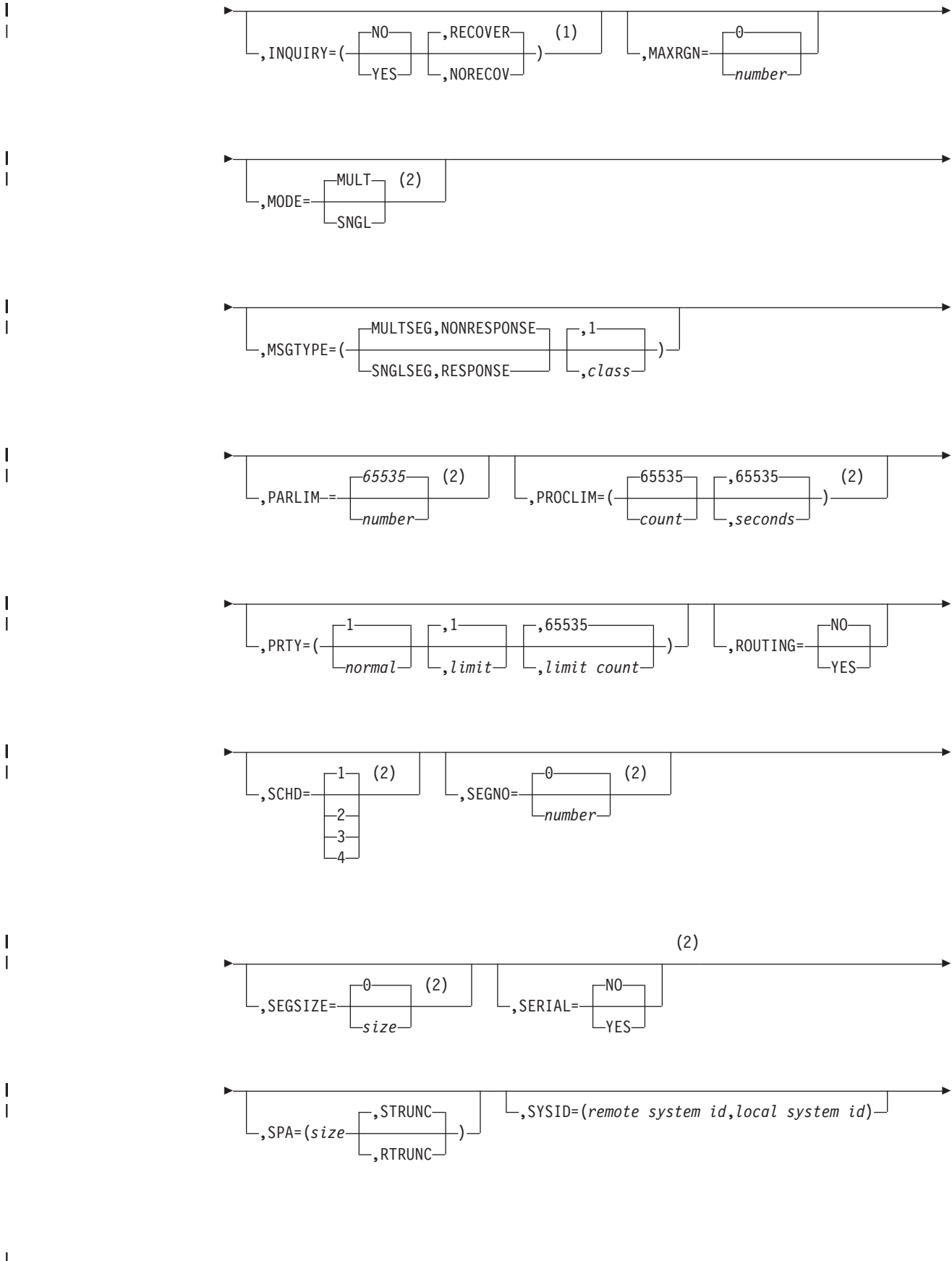
All Fast Path transactions are implicitly defined as recoverable.

MSGTYPE=(SNGLSEG,RESPONSE) must be specified on the TRANSACT statement for Fast Path potential transactions.

Additional routing codes can be associated with a Fast Path application program by using the RTCODE macro statement following the corresponding APPLCTN macro statement.

TRANSACT macro statements with FPATH=YES and RTCODE macro statements are invalid after APPLCTN macro statements that define an IMS batch message processing (BMP) application.





**Notes:**

- 1 NORECOV is not valid when NO is specified.
- 2 These operands are not appropriate if a transaction destined for processing in another (remote) system is being specified.

**Keyword Parameters**

To find which parameters apply to your IMS configuration, refer to Table 15 on page 33.

**AOI=**

**For Type 1 AOI:** For Type 1 Automated Operator Interface (AOI), specifies whether (YES | TRAN | CMD) or not (NO) a particular transaction is allowed to issue the AOI command (CMD) call.

**YES** When AOI=YES is specified, the authorization of the commands is similar to authorization for ICMD calls; the user ID or program name is used for authorization. For some environments, if a Get Unique call has not yet happened, then the program name is used for the authorization.

**TRAN** The TRAN specification is the same as YES, but also requests that IMS use the transaction code, instead of the user ID of the user who entered the transaction, to check the authorization of the commands for the CMD calls issued by the transaction. Use of the transaction code provides authorization checking that is similar to the authorization provided by SMU transaction-command security.

When a transaction is defined with AOI=TRAN, the first authorization check done for AOI for the transaction results in the security environment (ACEE) being built. This environment is kept for use by future authorization checks. In this case, the Type 1 AOI transactions need to be defined to RACF (or equivalent product) as a user. The transactions will then be specified on RACF PERMIT statements for each command they are allowed to issue from a Type 1 AOI transaction. Specifying AOI transactions as users to RACF might conflict with the name of a user already defined to RACF. If this occurs, then either the transaction name or the existing user name needs to be changed.

**CMD** The CMD specification is the same as YES, but also indicates that authorization checking is based on which transactions can issue a particular command. In this case, the commands (or the first three characters of the commands) need to be defined to RACF (or equivalent product) as a user. The Type 1 AOI transactions must be defined as profiles under the TMS class, and for each transaction, the commands it can issue must be specified. Using the CMD specification requires you to create fewer user IDs than you need to create for the AOI=TRAN specification. However, using the CMD specification requires you to create or modify a larger number of resource profiles.

Table 24 describes the relationship between the AOI= specifications on the TRANSACT macro and the specifications on the EXEC parameter AOI1=.

Table 24. Relationship Between AOI= and AOI1= Specifications

AOI1= specification	Affect on security	Affect on AOI= specifications
S	SMU is used for Type 1 AOI security, and IMS uses the transaction code to determine if a transaction is authorized to issue a particular command (the user ID is never used in this case).	YES specification is reset; TRAN and CMD specifications are retained.
R   C   A	SMU is not used.	All specifications are retained.
N	SMU is not used.	YES specification is reset; TRAN and CMD specifications are retained.

**For Type 2 AOI:** The AOI parameter is also available for Type 2 AOI security. However, AOI=N only applies to Type 1 AOI. If AOI=N is specified or set by default, transactions are allowed to use Type 2 AOI and issue the ICMD call.

If the TRAN specification is used for Type 2 AOI, IMS uses the transaction code, instead of the user ID of the user who entered the transaction, to check the authorization of the commands for the ICMD calls issued by the transaction. Using the transaction code provides authorization checking that is similar to the checking provided by SMU transaction-command security.

If the CMD specification is used for Type 2 AOI, authorization checking is based on which transactions can issue a particular command.

For IMS environments that do not support the use of a transaction, such as a non-message-driven BMP, the AOI=TRAN | CMD specification is not available, and the security checking that is performed is similar to the checking performed when YES is specified.

**CODE=**

Specifies one or more one- to eight-character alphanumeric transaction codes or remote transaction codes. If more than one code is specified, each transaction code name is assumed to have the same characteristics as all other keyword or positional parameter specifications or defaults. Enclose multiple names in parentheses and separate them with commas. Each character of transaction codes and logical terminal names must be an alphanumeric character (A through Z, #, \$, @, or 0 through 9). Transaction codes, logical terminals on the NAME macro, and linknames on MSNAME macros must comprise a set of values, each of which is unique in the system. That is, transaction codes, logical terminal names, and MSC linknames, collectively, cannot contain duplicates. The CODE operand is required.

**Example:**

CODE=(TRAN1,TRAN2,TRAN3)

**DCLWA=**

Specifies whether (YES) or not (NO) IMS should perform log write-ahead for recoverable, nonresponse mode input messages and transaction output messages. If not specified in the TRANSACT macro, the default is the DCLWA parameter in the IMSCTRL macro. The DCLWA parameter in the TRANSACT macro overrides the IMSCTRL macro parameter for this transaction.

Specify or accept a default of YES to ensure that both of the following occur:

- A nonresponse input transaction is made recoverable across IMS failures, prior to IMS acknowledging receipt of the input.

- Database changes are made recoverable prior to IMS sending associated output reply messages.

YES ensures that information in the log buffers is written to the IMS log, before the associated input acknowledgment or output reply is sent to the terminal.

Specify or accept a default of YES for all VTAM terminal types. For BTAM terminals, this option might increase input and output message path lengths.

**Related Reading:** For a information about the performance implications of selecting log write-ahead for transactions, see *IMS Version 9: Administration Guide: System*.

Specify NO if input message integrity and the consistency of output messages with associated database updates is not required. DCLWA does not apply to response mode or Fast Path input processing, and is ignored during IMS execution.

#### **EDIT=**

Specifies whether (UC) or not (ULC) the input data is to be translated to uppercase. The first parameter of this operand defines whether the transaction is uppercase/lowercase (ULC) as entered from the terminal, or if it is to be translated to uppercase (UC) before being presented to the processing program. The default is UC.

Specifying UC for VTAM terminals prevents the transmission of embedded device control characters.

You can also use EDIT to specify the one- to eight-character name of your own transaction input edit routine that edits messages prior to the program receiving the message. This name must begin with an alphabetic character. The specified edit routine (load module) must reside on the USERLIB data set prior to IMS system definition stage 2 execution. This routine cannot be the same as the one that is used on a LINEGRP or TYPE EDIT= parameter.

If FPATH=YES is specified, the EDIT= keyword parameter specifies whether (UC) or not (ULC) the transaction is to be translated to uppercase before being presented to the edit/routing exit routine. Specification of a user edit routine is valid on a Fast Path potential transaction; this specification is used when the transaction is routed to IMS. It is invalid on a Fast Path exclusive transaction.

For input from LU 6.2 devices, the user edit exit routine DFSLUEE0 is called instead of the transaction input edit routine specified in EDIT.

**Related Reading:** For more information about DFSLUEE0, see *IMS Version 9: Customization Guide*.

#### **FPATH=**

Specifies whether (YES, size) or not (NO) the transaction code is a potential candidate for Fast Path processing. FPATH=YES is effective only when specified on a TRANSACT statement that follows an APPLCTN statement that does **not** specify FPATH=YES; otherwise, the operand is ignored. FPATH=size, which determines the EMH buffer size required to run the transaction, overrides the EMHL execution parameter and implies FPATH=YES. The minimum specification for FPATH=size is 12; the maximum is 30720. The default is FPATH=NO.

Fast Path-potential transactions must be processed by a user edit/routing exit to determine whether the transaction is actually to be processed by IMS Fast Path. If it is to be processed by IMS Fast Path, the edit/routing exit routine

associates the transaction with a routing code. This routing code identifies which Fast Path application program is to process the transaction.

If a Fast Path–potential transaction is also defined with transaction command security by the Security Maintenance utility (SMU), that transaction must be processed by IMS, not Fast Path. (That is, an automated operator interface transaction can also be a Fast Path potential transaction if you program your user edit/routing exit routine to route the transaction to IMS and not Fast Path.) The SMU cannot define a Fast Path–exclusive transaction as able to issue commands.

If FPATH=YES is specified during a MODBLKS system definition, Fast Path must have been previously defined for the online system.

#### **INQ= or INQUIRY=**

Specifies whether (YES) or not (NO) this is an inquiry transaction. The default is NO. You can specify this keyword either as INQ= or INQUIRY=. If INQ= (or INQUIRY=) YES is specified, you can also specify whether (RECOVER) or not (NORECOV) this transaction should be recovered during an IMS emergency or normal restart. The specification of INQ=(NO,NORECOV) is invalid. The default is RECOVER.

For IMS Fast Path transactions, RECOVER must be specified.

INQ=YES should be specified only for those transactions that, when entered, do not cause a change in any database. Programs are prohibited from issuing ISRT, DLET, or REPL calls to a database when scheduled to process a transaction defined as INQ=YES. The INQ operands are not position dependent.

If the SPA parameter is specified (indicating that the transaction is conversational), INQ=(YES,NORECOV) cannot be specified.

Because switched terminals signed on for the INQUIRY LTERM, (one that has not specified /IAM) 2741 terminals are not allowed to enter transactions that update a database, the INQ=YES parameter must be specified for transactions that are entered by these terminals.

If an attempt is made to enter a transaction that is not specified as INQ=YES from one of these terminals, the transaction is rejected. A message indicating that the update transaction cannot be processed is sent to the terminal that entered the transaction.

#### **MAXRGN=**

Limits the number of message processing program (MPP) regions that can be concurrently scheduled to process a transaction. When the number of MPP regions is not limited, one transaction might monopolize all available regions.

If you specify zero, or if zero is defaulted to, no limit is imposed. The maximum number you can specify is 255.

If you specify SERIAL=YES or SCHDTYP=SERIAL in the APPLCTN macro, omit the MAXRGN parameter or set it equal to 0.

For non–0 values, you cannot specify MAXRGN= unless you also specify PARLIM=.

#### **MODE=**

Specifies that database buffers are to be written to direct access (flushed) upon

---

5. An application program cannot do an SQL INSERT, DELETE, or UPDATE when the IMS transaction is defined with INQ=YES.

each request for a new message (SNGL) by the processing program, or upon program termination (MULT). The default is MULT. Conversational and WFI transactions must be defined as SNGL. SNGL is forced for WFI applications.

This operand affects emergency restart. When MODE=SNGL, emergency restart only reprocesses the last completed message, regardless of whether one or more messages are scheduled and processed by a single load of an application program. Otherwise, emergency restart reprocesses those messages that were scheduled and processed by the single load of an application program, since the time of the previous checkpoint. The number of messages processed depends upon when the last checkpoint is issued.

The MODE= keyword parameters are checked on all IMS Fast Path–potential transactions for MODE=SNGL. If not specified, a warning diagnostic is issued.

### **MSGTYPE=**

Specifies the type of transaction code (single or multiple segment), and whether the communication line from which the transaction is entered is to be held until a response is received. The MSGTYPE operands are not position dependent.

The transaction code can be single segment (SNGLSEG), or multiple segment (MULTSEG). It specifies the time at which an incoming message is considered complete and available to be routed to an application program for subsequent processing. The defaults are (MULTSEG, NONRESPONSE, 1).

If MSC-directed routing is used in a multiple IMS system configuration, IMS does not ensure that both the message and the transaction destined to process that message are either single segment or multiple segments.

The MSGTYPE= keyword parameters are checked on all IMS Fast Path–potential transactions for MSGTYPE=(SNGLSEG, RESPONSE). If not specified, a warning diagnostic is issued.

The first parameter of the MSGTYPE keyword specifies one of the following choices for number of segments:

#### **MULTSEG**

Specifies that the incoming message can be more than one segment in length. It is not eligible for scheduling to an application program until an end-of-message indication is received, or a complete message is created by MFS.

#### **SNGLSEG**

Specifies that the incoming message is one segment in length. It becomes eligible for scheduling when the terminal operator indicates end-of-segment.

The second parameter of the MSGTYPE keyword specifies one of the following response choices:

#### **NONRESPONSE**

Specifies that, for terminals specifying or accepting a default of OPTIONS=TRANRESP, input should not stop after this transaction is entered.

#### **RESPONSE**

Specifies that, for terminals specifying or accepting a default of OPTIONS=TRANRESP, no additional messages are to be allowed after this transaction is entered until this transaction sends a response message back to the terminal. Response mode can be forced or negated by individual terminal definition.

| The third parameter of the MSGTYPE= keyword specifies the class to which  
| this transaction code is to be assigned. This parameter must be a decimal  
| number from 1 to 999. The default is 1.

Because the values associated with the class keyword can be changed using an /ASSIGN command, they will not be affected by an online change sequence of /MODIFY operator commands for an existing transaction, regardless of whether these values are altered during a MODBLKS system definition. The value specified must not exceed the MAXCLAS= value specified or accepted by default on the IMSCTRL macro statement. Regardless of specification made or default taken, remote transaction codes are assigned a class of zero. If the transaction code class is specified in the APPLCTN macro, this parameter need not be specified. If the transaction code class is specified in both the APPLCTN and TRANSACT macros, the APPLCTN macro specification is ignored for this transaction. Define CPI transactions with a different message class from that used for non-CPI transactions. IMS handles all CPI transactions as priority zero within the transaction class.

An input transaction from a 2741 line, or 2740 nonstation control is treated as a RESPONSE-type transaction, without regard to the type specified for this operand. Note also that MSGTYPE=RESPONSE is ignored during online processing for all terminals that do not operate in response mode.

#### **PARLIM=**

Specifies the threshold value to be used when SCHDTYP=PARALLEL is specified in the preceding APPLCTN macro instruction. If the current transaction enqueue count<sup>6</sup> exceeds the PARLIM value multiplied by the number of regions currently scheduled for this transaction, an additional region is scheduled upon a subsequent schedule event for this transaction's class. Events can include the following:

- Message Enqueue from terminal.
- Message insert from application. One region is allowed per syncpoint, EXPRESS=NO PCB. For example, when an application inserts two messages two messages for the same transaction code, only one region is scheduled at syncpoint.
- Assign command.
- Application term thread.

| If PARLIM is not specified, the default value of NONE is assumed, and IMS  
| allows the transaction to be scheduled in only one region at a time. Valid values  
| for PARLIM are from 0 to 32767.

PARLIM=0 indicates that any input message can cause a new region to be scheduled because the scheduling condition is always be met (the number of messages will be greater than zero).

If you specify PARLIM=0, you should specify a MAXRGN value to limit the number of regions that can be scheduled to process a particular transaction.

The value specified for PARLIM applies to message processing programs (MPPs) only; it is not supported for batch message processing programs (BMPs).

Because the value associated with the PARLIM= keyword can be changed using an /ASSIGN command, it is not affected by an online change sequence of

---

6. In a shared queues environment, the successful consecutive GU count is used instead of the enqueue count.



operator commands for an existing transaction, regardless of whether it is altered during a MODBLKS system definition.

If you specify SERIAL=YES or SCHDTYP=SERIAL in the APPLCTN macro, omit the PARLIM parameter and accept the default.

#### **PROCLIM=**

Specifies the number of messages (count) of this transaction code a program can process in a single scheduling, and the amount of time (in seconds) allowable to process a single transaction (or message). Batch Message Programs (BMPs) are not affected by these settings.

The first parameter, *count*, can be changed using an /ASSIGN command. Therefore, its value is not affected by an online change sequence of operator commands for an existing transaction, regardless of whether it was altered during a MODBLKS system definition.

The *count* specifies the maximum number of messages sent to the application program by the IMS control program for processing without reloading the application program. The *count* value can range from 0 through 65535. If 0 is coded, the maximum number of messages sent to the application is one and the application program is reloaded before receiving a subsequent message. Code the *count* value at 65535 if no limit is to be placed upon the number of messages processed at a single program load. Values 1 through 65535 are eligible for quick reschedule processing. The defaults for PROCLIM are 65535 and 65535.

The *second* parameter specifies a numeric value, in seconds, which can range from 1 to 65535.

If Fast Path is used, *second* specifies, for a given transaction code, the amount of time (in hundredths of seconds) the program is allowed to process a single transaction message. A valid specification ranges from 1 to 65535; if 65535 is specified, no time limit is placed on the application program. The time represents real time that elapses during transaction processing (not accumulated task time). Real time is used because the input terminal is in response mode and cannot enter another transaction until the response is sent. The *count* subparameter is ignored.

The number specified by *second* represents the maximum CPU time allowed for each message to be processed in the message processing region.

The *count* value assigned is used to determine how many messages an application program is allowed to process in a single scheduling cycle. When the application program requests, and receives, the number of messages indicated in the *count* value, any subsequent requests result in one of two things.

1. IMS indicates “no more messages exist” if any of the following conditions are true:

- The region is not an MPP.
- The currently scheduled mode is not MODE=SNGL.
- Equal or higher– priority transactions are enqueued for the region.

IMS might, in fact, have other messages enqueued for the application program. It is the responsibility of the application program to terminate when it receives an indicator that no more messages are available. Termination of the application program makes the region it occupied available for rescheduling. This feature makes it possible for IMS to allow scheduling of higher-priority transactions that entered the system while the previous

transactions were in process. In addition, if any equal-priority transactions are enqueued, they become eligible for scheduling on a first-in, first-out (FIFO) basis.

2. The region goes through quick reschedule and returns the next message to the application if all of the following conditions are true:
  - The region is an MPP.
  - The transaction is MODE=SNGL.
  - No higher priority transactions are enqueued.
  - Messages are still enqueued for the application.

The *seconds* value controls application program looping. You are not required to optimize the *seconds* value for program-transaction execution time. However, the *seconds* time value assigned should not be less than the expected per-transaction execution time. If the scheduled application program exceeds the product of *seconds* and *count*, the application program abends. If an IMS STIMER value of 2 is specified on the DFSMPR macro, the region does not abend until completion of the DL/I call.

The application must not use z/OS timer services, such as STIMER TASK, that override the IMS STIMER. IMS uses the IMS STIMER to time the execution of transactions. If an z/OS TIMER is issued, it disables the IMS STIMER.

#### **PRTY=**

Specifies the values that determine the scheduling priority of this transaction. This priority also controls the priority of messages created by this transaction and sent to a destination in a remote system.

##### **normal**

The priority assigned to this transaction when the number of input transactions enqueued and waiting to be processed is less than the *limit count* value. The valid specification range is from 0 through 14. The default is 1.

##### **limit**

The priority to which this transaction is raised when the number of input transactions enqueued and waiting to be processed is equal to or greater than the *limit count* value. The valid specification range is from 0 through 14. The default is 1.

##### **limit count**

The number that, when compared to the number of input transactions queued and waiting to be processed, determines whether the *normal* or *limit* priority value is assigned to this transaction. The *limit count* value can range from 1 through 65535. The default is 65535.

When the limit priority is used, and the priority is raised to the specified limit priority value, the priority is not reduced to the normal priority until all messages enqueued for this transaction code are processed.

If you do not want the limit priority for this transaction, code equal values for the normal and limit priorities, and a limit count of 65535.

When a transaction is processed exclusively by a batch message program (BMP), code the normal and limit priorities as 0. The limit count value is ignored for a transaction processed by a BMP.

The APPLCTN macro statement forces the scheduling priority of all transaction codes associated with it to 0 if the program type is batch (PGMTYPE=BATCH

on APPLCTN macro statement). However, a batch message processing region (BMP) can process transactions with scheduling priorities other than 0.

For remote transactions, the PRTY parameter determines the priority used to send the transaction to the processing system, which is termed the *MSC link message priority*. The three MSC link message priority groups are:

- Low
- Medium
- High

The low priority group consists of primary requests in the input terminal system. This group is assigned remote transaction priorities from 0 to 6. The medium group consists of secondary requests, responses, primary requests in an intermediate system, and primary requests in the input terminal system. This group is assigned a remote transaction priority of 7. The high group consists of primary requests in the input terminal system. Messages in this group are assigned remote transaction priorities from 8 to 14. Within each group, messages have a priority based on the current priority value of the transaction or remote transaction in the input terminal system for primary requests, and on the latest processing system for secondary requests and responses.

In an MSC configuration, the transaction priority determines the priority used to send messages inserted by this transaction across an MSC link. If the transaction inserts multiple messages to the same destination (for example, pages to a printer) and these messages must be sent in the order inserted, the normal and limit priority values should be the same. If the normal and limit priority values are not identical, messages inserted at a higher priority than previously inserted messages could arrive at their destination first. (This restriction does not apply to multiple segments of the same message.)

A transaction must have the same characteristics in all systems where it is defined. These characteristics include:

- Non-conversational/conversational
- SPA size if conversational
- Single-/multi-segment messages
- Non-inquiry/inquiry
- Recoverable/nonrecoverable

#### **ROUTING=**

If MSC directed routing is used in a multiple IMS system configuration, specifies whether (YES) or not (NO) the application program processing a transaction is informed of the system which originated the transaction.

If ROUTING=YES, an MSNAME corresponding to a logical path back to the originating system is placed in the I/O PCB. If ROUTING=NO, the name of the originating LTERM is placed in the I/O PCB. The default is NO.

#### **SCHD=**

Specifies the scheduling option used for other transactions when this transaction cannot be scheduled for internal reasons (database intent or no more space in PSB pool or DMB pool to bring in needed blocks). Values include:

- 1 Schedule only transactions of equal or higher priority in the selected class. This is the default.
- 2 Schedule higher-priority transactions in the selected class.

- 3 Schedule any transaction in the selected class.
- 4 Skip to the next class and attempt to schedule the highest-priority transaction in that class.

A loop counter in MPP scheduling breaks potential scheduling loops between SCHD=1 or SCHD=2 and SCHD=3 transactions. If scheduling fails due to intent conflicts for a SCHD=1 or SCHD=2 transaction, the counter increments and the next logical transaction is selected based on the SCHD=parameter of the failed transaction.

Each time the scheduler fails to schedule a transaction for intent, the counter is checked. If the count is greater than five, the next class of transaction is selected and the counter reset. This processing method can cause a delay in the current class.

For example, a long-running BMP has update intent on a database. Several transactions that have update intent on the same database with SCHD=1 are entered into the IMS system. Transactions in the same class that do not reference the database are also added to the system. However, the first group of transactions fails for intent, and none of the transactions is scheduled until the BMP terminates.

After each failed transaction, the loop counter increments. After the counter exceeds 5, the next class is scheduled, bypassing the transactions that do not reference the database and causing delays in their processing.

To avoid these delays, place the second group of transactions into a separate class, or run the BMP job at a different time.

**SEGNO=**

Specifies the maximum number of application program output segments that are allowed into the message queues per Get Unique (GU) call from the application program. It must be specified as a decimal number from 0 through 65535. The default is 0. If the default specification of 0 is used, the number of segments is not checked by the online system at execution time.

Because the value associated with the SEGNO= keyword can be changed using an /ASSIGN command, it is not affected by an online change sequence of operator commands for an existing transaction, regardless of whether it is altered during a MODBLKS system definition.

**SEGSIZE=**

Specifies the maximum number of bytes allowed in any one output segment. It must be specified as a decimal number from 0 through 65535. The default is 0. If the default specification of 0 is used, the segment size is not checked by the online system at execution time.

Because the value associated with the SEGSIZE= keyword can be changed using an /ASSIGN command, it is not affected by an online change sequence of operator commands for an existing transaction, regardless of whether it is altered during a MODBLKS system definition.

The maximum output message segment to a LU 6.2 device is 32767. If a transaction is expected to send output to a LU 6.2 device, the SEGSIZE parameter should be no greater than 32767. However, this is not enforced during processing of the TRANSACT macro, because IMS cannot determine the device type for the message destination until output time.

**SERIAL=**

Forces serial processing of messages for a given transaction. When SERIAL=YES, U3303 pseudoabends do not cause the message to be placed

on the suspend queue but rather on the front of the transaction message queue, and the transaction is stopped with a USTOP.

The USTOP of the transaction is removed when the transaction or the class is started with a /START command.

The default for this keyword is NO, which means message processing is done as before with the messages placed on the suspend queue after a U3303 pseudoabend. Scheduling continues until repeated failures result in the transaction being stopped with a USTOP.

If you specify SERIAL=YES, PARLIM= and MAXRGN= must equal 0 (you can omit these parameters). If PARLIM= or MAXRGN= is not 0, SERIAL=YES is invalid, and the default of NO is assumed.

### **SPA=**

Defines, by inclusion, that this transaction is a conversational transaction.

#### **size**

Specifies the size of the conversational scratchpad area (SPA). The size specified must be between 16 bytes and 32767 bytes inclusive.

### **STRUNC|RTRUNC**

You can turn the truncated data option on (STRUNC) or off (RTRUNC).

If you specify SPA=STRUNC, IMS preserves all of the data in the SPA, even when a program switch is made to a transaction that is defined with a smaller SPA. The transaction with the smaller SPA does not see the truncated data, but when the transaction switches to a transaction with a larger SPA, the truncated data is used.

If you specify SPA=RTRUNC, the truncated data is not preserved.

STRUNC is the default.

When a conversation initially starts, and when a program switches, the STRUNC|RTRUNC option is checked and set or reset as specified. When the option is set, it remains set for the life of the conversation, or until a program switch occurs to a transaction that specifies the option is to be reset.

When a program switch occurs, the truncated data option for the new transaction is first checked, and if specified (either STRUNC or RTRUNC), that specification is set for the conversation and is used for the SPA inserted into the output message. If the option is not specified for the new transaction, the option currently in effect for the conversation is used.

**Restriction:** The CORE, DASD, and FIXED operands can no longer be used. If you specify them, assembly errors occur.

### **SYSID=**

In the multiple-IMS system configuration, specifies the system identification (SYSID) of the remote system (the system on which the application executes) and the SYSID of the local system (the originating system to which the responses are returned). The values specified must be from 1 through 2036. The remote SYSID specified must also be defined in an MSNAME macro statement, but the local SYSID can be defined in any or all of the MSNAME, TRANSACT, and APPLCTN macro statements.

If the SYSID parameter is specified in the APPLCTN macro statement, you need not specify the SYSID in the TRANSACT macro statement. If the SYSID is specified in both the APPLCTN and the TRANSACT macro statements, the APPLCTN specification is ignored.

The SYSID parameter is independent of the link type (CTC, MTM, VTAM) specified on the TYPE= keyword of the MSPLINK macro statement. Because the values associated with the SYSID= keyword can be changed using an /MSASSIGN command, they are not affected by an online change sequence of operator commands, regardless of whether the values are altered during a MODBLKS system definition. The values will only take effect during an IMS cold start.

A Fast Path-exclusive transaction cannot have SYSID= specified. To assign a remote transaction as local, the associated APPLCTN macro must be defined as local. This means that no SYSID= parameter can be specified for the associated APPLCTN macro.

**Restriction:** A MODBLKS online change will be rejected if you attempt to add the SYSID parameter when a change to any other parameter specified on the TRANSACT macro is also requested.

**WFI=**

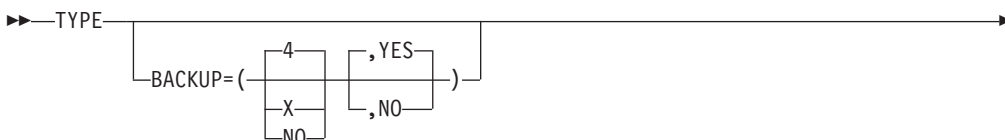
The positional parameter WFI specifies that this is a wait-for-input transaction. A message processing or batch processing application program that processes WFI transactions is scheduled and invoked normally. If the transaction to be processed is defined as WFI, the program is allowed to remain in main storage after it has processed the available input messages. The QC status code (no more messages) is returned to the program if the PROCLIM *count* is reached; a command is entered to change the status of the scheduled transaction, database, program or class; the /DBR, /DBD or /STA commands relating to the databases used by the transaction are entered, or IMS is terminated with a checkpoint shutdown. MODE=SNGL is forced when WFI is specified.

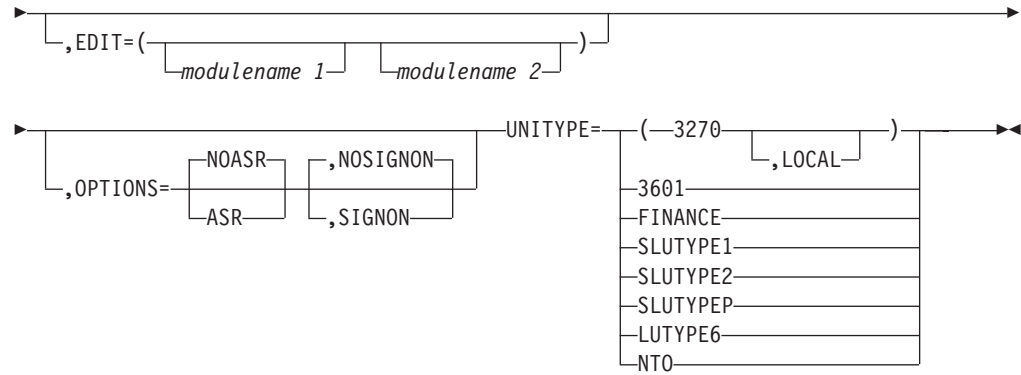
---

## TYPE Macro

The TYPE macro statement defines the beginning of a set of communication terminals and logical terminal description macro statements which include TERMINAL and NAME. The TYPE macro statement begins a description of one set, that contains one or more terminals of the same type. TYPE defines terminals attached to IMS through VTAM. It is equivalent to the LINEGRP/LINE macro set used to define terminals attached to IMS by means other than VTAM.

In addition to the keyword parameters described in this section, any keyword parameter of the TERMINAL macro statement except EDIT, LTERM, NAME, COMPT, COMPT1 through COMPT4, and ADDR can be specified as a keyword operand of the TYPE macro statement. This provides default values for subsequent TERMINAL macro statements within this terminal description set.





## Keyword Parameters

To find which parameters apply to your IMS configuration refer to Table 15 on page 33.

### BACKUP=

Specifies (for XRF only) the control of automatic restart (BTAM) or session switching (VTAM) after takeover. Use only when HSB=YES is specified on the IMSCTRL macro.

*X* is a numeric integer from 1 to 7, inclusive, that specifies priority for reestablishing the session. The default is 4 when either keyword or parameter is omitted. NO suppresses session recovery of the terminal at takeover.

Although IMS sets priorities for VTAM requests, the active requests can be completed in any order because of internal VTAM conflicts and pacing.

The second parameter determines whether the backup system is to attempt to establish a backup session to the terminal when the active request establishes a session. When the entire keyword or this parameter is omitted, the default is YES. Specifying NO suppresses the establishment of a backup session.

### EDIT=

*modulename 1* specifies the one- to eight-byte name of a user-supplied physical terminal output edit routine for the terminals in this communication description set.

*modulename 2* specifies the one- to eight-byte name of a user-supplied physical terminal input edit routine for the terminals in this communication description set.

This routine cannot be the same as the one that is used on a TRANSACT EDIT= parameter.

### OPTIONS=

- Specifies (for the Session Outage Notification facility only) automatic session restart (ASR) processing on all nodes of the same type. The default is NOASR.

The ASR and NOASR definitions on the TERMINAL macro override those definitions on the TYPE macro. However, the ASR option is not supported for nodes that can have XRF backup sessions.

Use the /DIS command to display the current ASR option, and the /CHANGE command to change it.

- Specifies signon verification security processing (SIGNON). The default is NOSIGNON.

The SIGNON and NOSIGNON definitions on the TERMINAL macro override those definitions on the TYPE macro.

Any signon option that is requested during system definition for a static terminal can also be requested in SMU. The SMU settings are applied during IMS restart.

Instead of specifying OPTIONS=SIGNON in more than one system definition macro, you can require all static terminals to sign on by specifying a single parameter, SIGNON=ALL, in the DFSDCxxx PROCLIB member.

**Related Reading:** For more information about the SIGNON keyword, see "Member DFSDCxxx" on page 338.

#### UNITYPE=

Specifies the terminal device type contained in this communication description set.

A terminal defined as either a 3770 or NTO is to be treated as a secondary logical unit type 1 terminal by IMS.

Terminals belonging to a Finance Communication System such as the 3600 Finance Communication System or the 4700 Finance Communication System can be specified as UNITYPE=FINANCE on the TYPE macro statement. Terminals belonging to the 3600 Finance Communication System can also be specified as UNITYPE=3601. The code generated to support the terminal is identical regardless of whether the terminal is specified as UNITYPE=3601 or UNITYPE=FINANCE.

---

## VTAMPOOL Macro

▶▶—VTAMPOOL—◀◀

This macro, required for parallel session support, begins the definition of the LU 6.1 LTERM subpools. You must define VTAMPOOL if any LU 6.1 node uses dynamic LTERM allocation. LTERMs, defined as LU 6.1 nodes, can be a fixed set or may be dynamically allocated depending on the composition of the VTAM macro set. This macro initiates a hierarchy of macros, consisting of a VTAMPOOL statement and one or more subsequent SUBPOOL macros, each of which can optionally be followed by one or more NAME macros (if at least one SUBPOOL statement has an associated NAME statement). This macro set can occur anywhere in the VTAM macro set if it does not break into the TYPE—TERMINAL—NAME statement hierarchy. This macro has no operands.

To find if this macro is valid in your IMS configuration, refer to Table 15 on page 33.

All LU 6.1 SUBPOOLS can be defined under a single VTAMPOOL statement or under multiple VTAMPOOL statements. The only purpose of defining LU 6 SUBPOOLS using multiple VTAMPOOL statements is to aid in documenting the use of each LU 6 SUBPOOL.



## Chapter 4. Procedures

This chapter provides the following information for each of the IMS System Definition-supplied procedures:

- A sample of each procedure
- Lists of valid parameters and DD statements for each procedure

The parameters are listed alphabetically at “Parameter Descriptions for IMS Supplied Procedures” on page 255. The DD statements are listed alphabetically at “DD Statement Descriptions” on page 303.

### **In this Chapter:**

- “Using IMS System Definition-Supplied Procedures” on page 193
- “Specifying the DFSRESLB DD Statement in IMS Batch Procedures” on page 194
- “DBBBATCH Procedure” on page 195
- “DBC Procedure” on page 198
- “DBRC Procedure” on page 203
- “DCC Procedure” on page 205
- “DFSJBP Procedure” on page 215
- “DFSJMP Procedure” on page 217
- “DFSMPR Procedure” on page 219
- “DLIBATCH Procedure” on page 220
- “DLISAS Procedure” on page 222
- “DXRJPROC Procedure” on page 225
- “FDR Procedure” on page 226
- “FPUTIL Procedure” on page 230
- “IMS Procedure” on page 231
- “IMSBATCH Procedure” on page 242
- “IMSCOBGO Procedure” on page 244
- “IMSCOBOL Procedure” on page 246
- “IMSFP Procedure” on page 248
- “IMSMSG Job” on page 249
- “IMSPLI Procedure” on page 249
- “IMSPLIGO Procedure” on page 250
- “IMSRDR Procedure” on page 252
- “RDIBATCH Procedure” on page 253
- Chapter 5, “Establishing a DL/I Interface from COBOL or PL/I,” on page 313
- “Parameter Descriptions for IMS Supplied Procedures” on page 255
- “DD Statement Descriptions” on page 303

Table 25 on page 192 shows all the IMS System Definition-supplied cataloged procedures, control statements, and jobs. The procedure is listed, followed by the environments in which they apply. Unless noted otherwise, all items in the first column of the table are procedures.

**Related Reading:** Refer to the *IMS Version 9: Administration Guide: System* for information about performance and tuning.

Table 25. IMS System Definition-Supplied Cataloged Procedures and the Environments in Which They Apply

Name	DB Batch	TM Batch	DBCTL	DB/DC	DBCTL	Description	Page
DBBBATCH	X	X				A one-step procedure for offline DL/I batch processing regions using IMS.ACBLIB.	195
DBC			X			An online execution procedure to initialize the DBCTL environment.	198
DBRC			X	X	X	A procedure to execute the DBRC address space.	203
DCC					X	An online execution procedure to initialize the DCCTL environment.	205
DFSJBP	X	X				Starts a JVM non-message driven dependent region, called a JBP region.	215
DFSJMP				X		Starts a JVM message-driven dependent region, called the JMP region.	217
DFSMR				X	X	A procedure to execute an IMS message processing address space.	219
DLIBATCH	X	X				A one-step procedure for an offline DL/I batch processing program using PSB and DBD libraries.	220
DLISAS			X	X		A procedure to execute the DL/I address space.	222
DXRJPROC	X		X	X		A procedure that defines the Internal Resource Lock Manager and its use to the system.	225
FDR			X	X		A procedure to execute an FDBR region.	226
FPUTIL			X	X		A procedure to execute the Fast Path utility program with DEDBs online.	230
IMS				X		An online execution procedure to initialize the DB/DC environment.	231
IMSBATCH			X	X	X	A procedure to execute an IMS online batch message processing address space.	242
IMSCOBGO	X	X				A three-step compile, bind, and go procedure combining the IMSCOBOL procedure with an execution step for a stand-alone DL/I batch address space.	244
CBLTDLI <sup>2</sup>	X	X	X	X	X	Control statements necessary to establish a COBOL-to-DL/I interface.	313
IMSCOBOL	X	X	X	X	X	A two-step compile and bind procedure for IMS applications written in COBOL.	246
IMSFP				X	X	A procedure for executing a Fast Path application program.	248
IMSMSG <sup>1</sup>				X	X	A job to execute an IMS message processing program.	249
IMSPLI	X	X	X	X	X	A two-step compile and bind procedure for IMS applications written in PL/I.	249

Table 25. IMS System Definition-Supplied Cataloged Procedures and the Environments in Which They Apply (continued)

Name	DB Batch	TM Batch	DBCTL	DB/DC	DBCTL	Description	Page
IMSPLIGO	X	X				A three-step compile, bind, and go procedure combining the IMSPLI procedure with an execution step for a stand-alone DL/I batch processing address space.	250
IMSRDR			X	X	X	A DASD read procedure to read an IMSMSG job into the operating system job stream from direct access storage devices.	252
PLITDLI <sup>2</sup>	X	X	X	X	X	Control statements necessary to establish a PL/I-to-DL/I interface.	313
RDIBATCH	X		X	X		A procedure to help maintain database availability in failure situations.	253

**Notes:**<sup>1</sup> Job<sup>2</sup> Control Statement

**Related Reading:** The information concerning the following supplied cataloged procedures, control statements, and jobs is found in the following publications:

*IMS Version 9: Utilities Reference: Database and Transaction Manager*      *IMS Version 9: Utilities Reference: System*

DFSMREC	ACBGEN
DFSWTnnn	DBDGEN
FMTCPY	DFSACBCP
IMSMSV	DFSMREC
IMSWTnnn	IMSDALOC
INITMOD	PSBGEN
MFSBACK	SECURITY
MFSBTCH1	
MFSBTCH2	
MFSDCT	
MFSREST	
MFSRVC	
MFSTEST	
MFSUTL	
OLCUTL	
REFCPY	

## Using IMS System Definition-Supplied Procedures

This section describes some general considerations on creating, modifying, and storing IMS system definition-supplied procedures.

### Storing Procedures

The DBC, DBRC, DCC, DLISAS, IMS, and IMSRDR procedures should be stored in SYS1.PROCLIB. The JCL supplied in the IMSMSG and IMSWTnnn members of

IMS.PROCLIB must be modified as your installation requires and stored in IMS.JOBS. The other IMS-supplied procedures can be left in IMS.PROCLIB, where they are placed during IMS system definition.

If PROCLIB=YES is specified when preparing the IMSGEN system definition macro statement, certain procedures and the jobs IMSMSG and IMSWTnnn are dynamically created and placed in IMS.PROCLIB. The created jobs and procedures should be examined carefully to determine whether the JCL is generated as you require. These procedures might not apply to all applications, but they can be used as guidelines for user-generated account oriented procedures. Depending on the type of system being defined, your procedure library members can be a subset of the complete IMS procedure library that is presented here.

IMS conforms to z/OS rules for data set authorization. If an IMS job step is to run authorized, all libraries to be used in that job step must be authorized. To run an IMS batch region as unauthorized, a nonauthorized library must be concatenated to IMS.SDFSRESL.

## Allocating Terminal Devices in an Online IMS System

If an online IMS system has been defined, particular attention should be devoted to the terminal device allocation generated within the IMS procedure. A list of terminal addresses and logical and physical terminals is printed by stage 1 of IMS system definition. Examples of the procedure jobs in this chapter show the contents of the members as they are supplied. No card column image is intended. When coding your own procedures, follow JCL and Assembler language coding practices.

## Specifying the DFSRESLB DD Statement in IMS Batch Procedures

Certain batch procedures must load the IMS SVC modules, all of which must reside in an authorized library. The DFSRESLB DD statement specifies the library that contains these modules. All libraries specified by this statement must be authorized through the Authorized Program Facility (APF).

When you include the DFSRESLB DD statement in a batch procedure, all IMS SVC modules are loaded from the specified library. If this library does not contain the required modules, IMS does not search any other library for the modules and then terminates.

When you do not include the DFSRESLB DD statement in a batch procedure, IMS searches the standard default libraries for the modules (JOBLIB/STEPLIB, followed by LINKLIB). If the IMS SVC modules do not reside in an authorized library, IMS terminates.

In addition to the DFSRESLB DD statement, the EXEC parameter SRCH, which is allowed on selected batch procedures, also affects the library search order. When SRCH=1, IMS searches the link pack area and directories prior to searching any other library. When SRCH=0 and the DFSRESLB DD statement is specified, IMS searches only the specified library and does not search the link pack area.

The following modules must be loaded from an authorized library during the IMS batch initialization:

DFSABND0	DFSMODU0
DFSAOSF0	DFSRTM00
DFSAOS70	DFSSPF00
DFSBC000	DFSSTM00

DFSCBT10  
DFSFSWA0  
DFSKPXT0

DFSTOPR0  
DFSVC100  
DFSVC110

---

## DBBBATCH Procedure

The DBBBATCH procedure is a one-step procedure for running an offline DL/I batch processing region using IMS.ACBLIB.

The following restrictions apply when you run DL/I batch jobs in an RSR environment:

### **Restrictions:**

- The batch-only databases that are identified to be tracked must be included in the system definition (DATABASE macro) for the tracking subsystem.
- The default GSG and TMI names used by batch jobs are specified in the IMSCTRL macro, but they can be overridden in the DLIBATCH procedure.
- Batch jobs do not use the DFSSRxx PROCLIB member, and are therefore limited to the default VTAM modename (TMDEFLT).

## DBBBATCH Procedure Description

The procedure shown in Figure 15 on page 196 runs an offline DL/I batch processing region using IMS.ACBLIB.

If VSAM databases are used, see “Specifying IMS Buffer Pools” on page 430.

```

|          //          PROC MBR=TEMPNAME,PSB=,BUF=7,
|          //          SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,
|          //          SRCH=0,CKPTID=,MON=N,LOGA=0,FMTO=T,
|          //          IMSID=,SWAP=,DBRC=,IRLM=,IRLMNM=,
|          //          BKO=N,IOB=,SSM=,APARM=,
|          //          RGN=4M,
|          //          SOUT=A,LOGT=2400,
|          //          SYS=,
|          //          SYS2=,LOCKMAX=,
|          //          GSGNAME=,TMINAME=,RRS=N,
|          //          IMSPLEX=
|          //G        EXEC PGM=DFSRR00,REGION=&RGN,
|          //          PARM=(DBB,&MBR,&PSB,&BUF,
|          //          &SPIE&TEST&EXCPVR&RST,&PRLD,
|          //          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,
|          //          &IMSID,&SWAP,&DBRC,&IRLM,&IRLMNM,
|          //          &BKO,&IOB,&SSM,
|          //          '&APARM',&LOCKMAX,
|          //          &GSGNAME,&TMINAME,&RRS,
|          //          &IMSPLEX)
|          //STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
|          //          DD DSN=IMS.&SYS2.PGMLIB,DISP=SHR
|          //DFSRESLB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
|          //IMSACBA DD DSN=IMS.&SYS2.ACBLIBA,DISP=SHR
|          //IMSACBB DD DSN=IMS.&SYS2.ACBLIBB,DISP=SHR
|          //MODSTAT DD DSN=IMS.&SYS2.MODSTAT,DISP=SHR
|          //MODSTAT2 DD DSN=IMS.&SYS2.MODSTAT2,DISP=SHR
|          //PROCLIB DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
|          //IEFRDER DD DSN=IMSLOG,DISP=(,KEEP),VOL=(,,99),
|          //          UNIT=(&LOGT,,DEFER),
|          //          DCB=(RECFM=VB,BLKSIZE=4096,
|          //          LRECL=4092,BUFNO=2)
|          //IEFRDER2 DD DSN=IMSLOG2,DISP=(,KEEP),VOL=(,,99),
|          //          UNIT=(&LOGT,,DEFER,SEP=IEFRDER),
|          //          DCB=(RECFM=VB,BLKSIZE=4096,
|          //          LRECL=4092,BUFNO=2)
|          //SYSUDUMP DD SYSOUT=&SOUT,
|          //          DCB=(RECFM=FBA,LRECL=121,BLKSIZE=605),
|          //          SPACE=(605,(500,500),RLSE,,ROUND)
|          //IMSMON DD DUMMY

```

Figure 15. Procedure to Run Offline DL/I Batch Processing Region Using IMS.ACBLIB.

#### Notes:

- Parameters in parentheses are positional.
- For an XRF-capable system only.
- The IEFRDER statement is not required if the job does not declare database update intent.
- IEFRDER2 is optional.

For a job step declaring database update intent, DD DUMMY can be specified, if the job step is not using DBRC. This is valid where an image copy of the database is taken before the update job step.

Log initialization calculates the smallest value necessary for logical record length. If the JCL logical record length value is larger than the calculated value, the JCL value is used; otherwise, log initialization uses the calculated value for logical record length and adds 4 for the block size.

If multiple volumes are required for the system log, a volume count value should be specified in the VOL parameter of the DD statement.

**Related Reading:** For information on the number of volumes required per DD statement, see *MVS/ESA Job Control Language User's Guide*.

If the IBM 3480 tape drive is used for the IMS log data set, IMS forces tape write mode (DCB=OPTCD=W). The default on the 3480 is to buffer the write so that IMS cannot detect when the write is actually performed. If a power failure occurs after a log record is written to the 3480, and the database is updated but the log record is not yet written to tape, database integrity is lost. Tape write mode is forced for the log in batch and GSAM data sets.

## DBBBATCH Parameters

Table 26 lists the parameters that are valid for the DBBBATCH procedure. See "Parameter Descriptions for IMS Supplied Procedures" on page 255.

You can specify these parameters in the PARM1= and PARM2= parameters.

*Table 26. Valid parameters for the DBBBATCH procedure*

APARM=	MON=
BKO=	PRLD=
BUF=	PSB= <sup>2</sup>
CKPTID=	RGN=
DBRC=	RGSUF= <sup>4</sup>
EXCPVR= <sup>3</sup>	RST= <sup>3</sup>
FMTO=	SOUT=
GSGNAME=	SPIE= <sup>3</sup>
IMSID=	SRCH=
IOB= <sup>1</sup>	SSM=
IRLM=	SWAP=
IRLMNM=	SYS=
IMSPLEX=	SYS2=
LOCKMAX=	TEST= <sup>3</sup>
MBR= <sup>3</sup>	TMINAME=

### Notes:

1. This parameter is no longer used and is ignored if specified.
2. This parameter is optional.
3. This parameter cannot be specified in the PARM1= and PARM2= parameters.
4. There is no default when specified in the batch procedure.

## DBBBATCH DD Statements

The following DD statements are valid for the DBBBATCH procedure.

In addition to the following DD statements, add statements for data sets representing IMS databases that are not to be dynamically allocated.

For explanations of the DD statements, see "DD Statement Descriptions" on page 303.

DFSHALDB DD <sup>1</sup>	IMS LOGR DD
DFSRESLB DD	MODSTAT DD
DFSSTAT DD <sup>1</sup>	MODSTAT2 DD
DFSVSAMP DD	OLCSTAT DD
IEFRDER DD	PROCLIB DD
IEFRDER2 DD <sup>1</sup>	RECON <sub>n</sub> DD <sup>1</sup>
IMSACB DD	STEPLIB DD
IMSACBA DD	SYSHALDB DD <sup>1</sup>
IMSACBB DD	SYSUDUMP DD
IMS DD	

**Note:**<sup>1</sup> This statement is optional.

## DBC Procedure

The DBC procedure is an online execution procedure to initialize the DBCTL environment.

## DBC Procedure Description

The procedure shown in the following example executes an IMS DBCTL control region.

```

//      PROC RGN=64M,SOUT=A,DPTY='(14,15)',
//      SYS=,SYS2=,
//      LOGT=2400,SYS1=,
//      RGSUF=DBC,PARM1=,PARM2=
//IEFPROC EXEC PGM=DFSMVRC0,DPRTY=&DPTY,REGION=&RGN,
//      PARM='DBC,&RGSUF,&PARM1,&PARM2'
//*
//*
//* THE MEANING AND MAXIMUM SIZE OF EACH PARAMETER
//* IS AS FOLLOWS:
//*****
//*   RGSUF   XXX   EXEC PARM DEFAULT BLOCK SUFFIX FOR
//*             MEMBER DFSPBXXX.
//*****
//*
//*   PARM1 , PARM2 PARAMETERS BOTH ARE USED TO SPECIFY
//*   CHARACTER STRINGS THAT CONTAIN IMS KEYWORD
//*   PARAMETERS. I.E. PARM1='AUTO=Y,PST=222,RES=Y'
//*
//* ALL OF THE VALID DBCTL KEYWORD PARAMETERS
//* ARE DESCRIBED BELOW
//*****
//***** CONTROL REGION SPECIFICATIONS *****
//* RES     X     BLOCK RESIDENT (N = NO, Y = YES)
//* PST     XXX   NUMBER OF PST'S PERMANENTLY ALLOC
//* MAXPST XXX   MAXIMUM NUMBER OF PST'S
//* SRCH    X     MODULE SEARCH INDICATOR FOR DIRECTED LOAD
//*             0 = STANDARD SEARCH
//*             1 = SEARCH JPA AND LPA BEFORE PDS
//* FMTO     T = ONLINE FORMATTED DUMP WITH
//*             STORAGE IMAGE DELETIONS.
//*             TERM=NO SDUMPS ALLOWED.
//*             P = FULL ONLINE FORMATTED DUMP.
//*             TERM=NO SDUMPS ALLOWED.
//*             F = FULL ONLINE FORMATTED DUMP.
//*             TERM=NO SDUMPS NOT ALLOWED.

```



```

|          /**          N = NO FORMATTED DUMP, NO OFFLINE
|          /**          DUMP. TERM=NO SDUMPS ALLOWED.
|          /**          Z = NO FORMATTED DUMP, NO OFFLINE
|          /**          DUMP. TERM=NO SDUMPS NOT
|          /**          ALLOWED.
|          /** (DEFAULT) D = OFFLINE DUMP, OR ONLINE FORMAT-
|          /**          TED DUMP WITH STORAGE IMAGE
|          /**          DELETIONS IF OFFLINE DUMPING
|          /**          FAILS. TERM=NO SDUMPS ALLOWED.
|          /**          X = OFFLINE DUMP, OR ONLINE FORMAT-
|          /**          TED DUMP WITH STORAGE IMAGE
|          /**          DELETIONS IF OFFLINE DUMPING
|          /**          FAILS. TERM=NO SDUMPS NOT
|          /**          ALLOWED.
|          /**          M = OFFLINE DUMP, ONLINE IMS DUMP
|          /**          FORMATTING NOT PERMITTED
|          /**          TERM=NO SDUMPS ALLOWED.
|          /**          R = OFFLINE DUMP, ONLINE IMS DUMP
|          /**          FORMATTING NOT PERMITTED
|          /**          TERM=NO SDUMPS NOT ALLOWED.
|          /** AUTO X Y = AUTOMATIC RESTART DESIRED
|          /**          N = NO AUTOMATIC RESTART
|          /** IMSID XXXX IMS SUBSYSTEM IDENTIFIER
|          /** ISIS X N = NO RESOURCE ACCESS SECURITY
|          /**          R = RACF RESOURCE ACCESS SECURITY
|          /**          C = RACF RESOURCE ACCESS SECURITY
|          /**          A = RACF RESOURCE ACCESS SECURITY
|          /**          0 = NO AGN SECURITY
|          /**          1 = APPLICATION GROUP NAME SECURITY
|          /**          2 = APPLICATION GROUP NAME SECURITY
|          /** ARMRST X Y = ALLOW MVS ARM TO RESTART
|          /**          N = ARM NOT RESTART IMS
|          /** RRS X Y = ENABLE PROT CONV SUPPORT
|          /**          N = DISABLE PROT CONV SUPPORT
|          /** IRLM X Y = YES, N = NO
|          /** IRLMNM XXXX IRLM SUBSYSTEM NAME
|          /** SSM XXXX EXT SUBSYSTEM PROCLIB MEMBER ID
|          /** WADS X SINGLE OR DUAL WADS,S=SINGLE,D=DUAL
|          /** ARC XX AUTOMATIC ARCHIVE.
|          /**          0 = NOT AUTOMATIC
|          /**          1-99 = AUTOMATIC
|          /** UHASH XXXXXXXX USER HASH MODULE NAME
|          /** DBRCNM XXXXXXXX DBRC PROCLIB MEMBER NAME
|          /** DLINM XXXXXXXX DL/I PROCLIB MEMBER NAME
|          /** PRDR XXXXXXXX IMSRDR PROCLIB MEMBER NAME
|          /** DBRSE XXXXXXXX 8 CHAR DBCTL RSENAME
|          /** CRC X COMMAND RECOGNITION CHARACTER
|          /** CMDMCS X COMMAND SECURITY OPTION
|          /**          R=RACF COMMAND SECURITY
|          /**          C=DFSCCMD0 COMMAND SECURITY
|          /**          B=RACF AND DFSCCMD0 CMD SEC
|          /** PREMSG X PREFIX MESSAGE OPTION
|          /**          N=NO DFS000I PREFIX
|          /**          Y=DEFAULT, DFS000I PREFIX
|          /** PIMAX XXXXXX ENQ/DEQ POOL MAXIMUM BYTES
|          /** PIINCR XXXXXX ENQ/DEQ POOL INCREMENT
|          /** AOIS X ICMO SECURITY OPTION
|          /** YEAR4 X N = 2-DIGIT DATE (DEFAULT)
|          /**          Y = 4-DIGIT DATE
|          /** CPLOG XXXXXX CHECKPOINT LOG INTERVAL
|          /** OR

```

```

|          /** CPLOG  XXM  CHECKPOINT LOG INTERVAL
|          /** IMSGROUP XXXX  4 CHAR USER SPEC NAME
|          /** DESC   XX   MSG DESC CODE
|          /** MCS    (XX,XX) MSG ROUTE CODES
|          /** SVC2   XXX   TYPE 2 SVC NUMBER
|          /**
|          /******* FDR PARAMETER *****
|          /**
|          /** FDRMBR XX   SUFFIX FOR FDR MEMBER IN
|          /**                IMS.PROCLIB
|          /**
|          /******* FAST PATH PARAMETERS *****
|          /**
|          /** BSIZ  XXX  DATA BASE BUFFER SIZE
|          /** OTHR  XXX  NUMBER OF OUTPUT THREADS
|          /** DBFX  XXX  SYSTEM ALLOCATION OF DATA BASE BUFFERS TO BE
|          /**                FIXED AT START OF 1ST FAST PATH DEP REGION
|          /** DBBF  XXX  NUMBER OF DATABASE BUFFERS
|          /** DBFP  XXXX  PAGE FIX/FREE ADJUST TIMER
|          /**                0: FIX/FREE AT SCHED/TERM
|          /**                1: ALLOW PAGEFIX ONLY
|          /**                2-3600: SEC PAGEFREE FREQ
|          /** LGNR  XX   NUMBER OF LOG ENTRIES IN DEDB BUFFERHEADER
|          /** SVSODR XXXX  SVSO DISASTER RECOVERY OPTIONS
|          /**                NONE: DEFAULT. NO CHANGE TO ERE.
|          /**                AUTO: AREA MARKED RECOV NEEDED IF
|          /**
|          /**                DRRS: AREA MARKED RECOV NEEDED AT
|          /**
|          /**                WTOR: USER OPTION TO MARK AREAS
|          /**
|          /** DMHVF XX  MEGS TO FIX PAGEFIX FOR VSO ERE      DATASPACE
|          /** FPOPX  X  DISABLE CTL REG PREOPEN FOR DEDBS
|          /**                BLANK. DEFAULT. PREOPEN OF
|          /**                DEDB AREAS DONE AT CTL REGION
|          /**                INITIALIZATION
|          /**                D: DEDB PREOPEN/PRELOAD IS DIS-
|          /**                ABLED AT CTL REGION INIT
|          /******* RSR PARAMETERS *****
|          /**
|          /** RSRMBR XX   SUFFIX FOR RSR MEMBER
|          /** TRACK  XXX  NO = NO RECOVERY TRACKING DONE
|          /**                RLT = RECOVERY TRACKING DONE
|          /**                DLT = DATABASE TRACKING DONE
|          /**
|          /******* STORAGE POOL VALUES IN K, M OR G *****
|          /** DMB   XXXXXX DMB POOL SIZE
|          /** CIOP  XXXXXX CIOP POOL UPPER LIMIT
|          /** WKAP  XXXXXX WORKING STORAGE BUFFER POOL SIZE
|          /** PSBW  XXXXXX PSB WORK POOL SIZE
|          /** DBWP  XXXXXX DATABASE WORK POOL SIZE
|          /** CSAPSB XXXXXX DLISAS: CSA PSB POOL SIZE
|          /** DLIPSB XXXXXX DLISAS: DLI PSB POOL SIZE
|          /** EPCB  XXXXXX EPCB POOL SIZE
|          /** FPWP  XXXXXX FP WORK POOL UPPER LIMIT
|          /** AOIP  XXXXXX AOI POOL UPPER LIMIT
|          /**
|          /******* MEMBER SUFFIXES *****
|          /**
|          /** SUF   X    LAST CHARACTER OF CTL PROGRAM LOAD
|          /**                MODULE MEMBER NAME

```

```

|          /* FIX    XX    2 CHARACTER FIX PROCEDURE MODULE SUFFIX
|          /* PRLD   XX    2 CHARACTER PROCLIB MEMBER SUFFIX FOR PRELOAD
|          /* VSPEC  XX    2 CHARACTER BUFFER POOL SPEC MODULE SUFFIX
|          /* SPM    XX    STORAGE POOL OPTIONS (DFSSPMXX)
|          /* CSLG   XXX   CSL GLOBAL MEMBER (DFSCGXXX)
|          /******
|          /*
|          /*STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
|          /*PROCLIB DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
|          /*
|          /******
|          /* IN ORDER TO START A DEPENDENT REGION, MODIFIED
|          /* START-UP JCL IS WRITTEN FROM INTERNAL STORAGE TO
|          /* THE INTERNAL READER.
|          /*
|          /*IMSIRD  DD SYSOUT=(A,INTRDR)
|          /*
|          /****** DASD LOGGING DD CARDS *****
|          /* THE FOLLOWING DD CARDS DESCRIBE THE DASD LOGGING
|          /* OLDS AND WADS. THESE CARDS ARE FOR EXAMPLE ONLY.
|          /* ALL OLDS AND WADS DATA SETS MAY BE DYNAMICALLY
|          /* ALLOCATED. DD CARDS ARE NOT REQUIRED.
|          /* THE OLDS AND WADS TO BE USED DURING STARTUP MUST
|          /* BE SPECIFIED VIA OLDSDEF AND WADSDEF CONTROL
|          /* STATEMENTS IN THE DFSVSMXX MEMBER OF IMS PROCLIB.
|          /* THE ACTUAL SELECTION OF OLDS AND WADS MUST BE
|          /* TAILORED TO INSTALLATION REQUIREMENTS. THE OLDS
|          /* AND WADS MUST BE PREDEFINED BY A SET UP JOB.
|          /* THE BLOCK SIZE OF ALL OLDS MUST BE THE SAME.
|          /* THE BLOCK SIZE AND DEVICE TYPE OF ALL WADS MUST
|          /* BE THE SAME. AT LEAST 3 PRIMARY OLDS AND 1 WADS
|          /* MUST BE AVAILABLE FOR STARTUP. THE BLOCK SIZE
|          /* SHOULD NOT BE SPECIFIED IN THIS JCL. THE LOGGER
|          /* WILL GET THE BLOCK SIZE FROM THE VTOC.
|          /*
|          /*DFSOLP00 DD DSN=IMS.&SYS.OLP00,DISP=SHR
|          /*DFSOLP01 DD DSN=IMS.&SYS.OLP01,DISP=SHR
|          /*DFSOLP02 DD DSN=IMS.&SYS.OLP02,DISP=SHR
|          /*DFSOLP03 DD DSN=IMS.&SYS.OLP03,DISP=SHR
|          /*DFSOLP04 DD DSN=IMS.&SYS.OLP04,DISP=SHR
|          /*DFSOLP05 DD DSN=IMS.&SYS.OLP05,DISP=SHR
|          /*
|          /*DFSOLS00 DD DSN=IMS.&SYS.OLS00,DISP=SHR
|          /*DFSOLS01 DD DSN=IMS.&SYS.OLS01,DISP=SHR
|          /*DFSOLS02 DD DSN=IMS.&SYS.OLS02,DISP=SHR
|          /*DFSOLS03 DD DSN=IMS.&SYS.OLS03,DISP=SHR
|          /*DFSOLS04 DD DSN=IMS.&SYS.OLS04,DISP=SHR
|          /*DFSOLS05 DD DSN=IMS.&SYS.OLS05,DISP=SHR
|          /*
|          /*DFSWADS0 DD DSN=IMS.&SYS.WADS0,DISP=SHR
|          /*DFSWADS1 DD DSN=IMS.&SYS.WADS1,DISP=SHR
|          /****** IMSMON DD CARDS *****
|          /*
|          /* THE IMSMON DD STATEMENT MUST BE REMOVED IF
|          /* THIS DATA SET IS TO BE DYNAMICALLY ALLOCATED.
|          /*
|          /*IMSMON  DD DSN=IMS.&SYS1.IMSMON,DISP=(,KEEP),
|          /*          VOL=(,.,99),UNIT=(&LOGT,.,DEFER)
|          /*
|          /*IMSACBA DD DSN=IMS.&SYS2.ACBLIBA,DISP=SHR
|          /*IMSACBB DD DSN=IMS.&SYS2.ACBLIBB,DISP=SHR

```

```

| //MODBLKSA DD DSN=IMS.&SYS2.MODBLKSA,DISP=SHR
| //MODBLKSB DD DSN=IMS.&SYS2.MODBLKSB,DISP=SHR
| //MODSTAT DD DSN=IMS.&SYS2.MODSTAT,DISP=SHR
| //***** SYSTEM REQUIRED DD CARDS *****
| //*
| //SYSUDUMP DD SYSOUT=&SOUT,
| //          DCB=(LRECL=125,RECFM=FBA,BLKSIZE=3129),
| //          SPACE=(6050,300,,ROUND)
| //IMSRDS DD DSN=IMS.&SYS2.RDS,DISP=SHR
| //MATRIXA DD DSN=IMS.&SYS2.MATRIXA,DISP=SHR
| //MATRIXB DD DSN=IMS.&SYS2.MATRIXB,DISP=SHR
| //PRINTDD DD SYSOUT=&SOUT
| //*
| //***** EXTERNAL SUBSYSTEM DD CARDS *****
| //*
| //* USER MAY OPTIONALLY ADD THE DFSESL DD CARD
| //* FOR EXTERNAL SUBSYSTEM CONNECTION.
| //*
| //***** DATA BASE DD CARDS *****
| //*
| //* USER MAY OPTIONALLY SUPPLY THE DD STATEMENTS
| //* FOR THE ON-LINE DATA BASES TO BE
| //* INSERTED HERE PRIOR TO ATTEMPTING
| //* AN ON-LINE SYSTEM EXECUTION USING
| //* THIS PROCEDURE.
| //* IF NO DD STATEMENTS ARE SUPPLIED FOR
| //* A DATA BASE, IMS ASSUMES THAT THIS
| //* DATA BASE HAS BEEN DESCRIBED THROUGH
| //* THE DFSMDA MACRO.
| //* IF THE USER WILL BE EXECUTING WITH THE DL/I
| //* SAS OPTION, THESE DD STATEMENTS SHOULD BE ADDED
| //* TO THE DLISAS PROCLIB MEMBER OR DESCRIBED
| //* THROUGH THE DFSMDA MACRO.

```

If DL/I databases are used, a member DFSVSMxx is built in IMS.PROCLIB. See “Specifying IMS Buffer Pools” on page 430. **Note:** DEDB areas must be allocated in the DBCTL control region. One of the following must occur:

- The DD statements be in the control region JCL.
- The dynamic allocation members must be in a library, as addressed by JOBLIB and STEPLIB.

DBCTL execution parameters can be specified on the DBC EXECUTE statement, in the DBC procedure, in the defaults member DFSPBDBC, or the parameter default values specified during system definition can be used. Following is a complete list of the parameters that apply to a DBCTL DBC region.

The DBC procedure is used “as is” for both regions in a DBCTL stand-by configuration.

## DBC Parameters

The following parameters are valid for the DBC procedure. Also, see “Parameter Descriptions for IMS Supplied Procedures” on page 255.

	AOIP=	EPCB=	PRDR=
	AOIS=	FDRMBR=	PREMSG=
	ARC=	FIX=	PRLD=
	ARMRST=	FMTO=	PSBW=
	AUTO=	FPDSSIZE=	PST=
	BSIZ=	FPOPN=	RES=
	CCTCVCAN=	FPWP=	RGSUF=
	CIOP=	IMSGROUP=	RRS=
	CMDMCS=	IMSID=	RSRMBR=
	CRC=	IOVFI=	SPM=
	CSAPSB=	IRLM=	SRCH=
	CSLG=	IRLMNM=	SSM=
	DBBF=	ISIS=	SUF=
	DBFP=	LGNR=	SVC2=
	DBFX=	MAXPST=	SVSODR=
	DBRCNM=	MCS=	TRACK=
	DBRSE=	ODBASE=	TSR=
	DBWP=	ORSMBR=	UHASH=
	DESC=	OTHR=	VSPEC=
	DLINM=	PIINCR=	WADS=
	DLIPSB=	PIMAX=	WKAP=
	DMB=		YEAR4=
	DMHVF=		

## DBC DD Statements

Valid DD statements for the DBC procedure are listed in this section.

In addition to the following DD statements, add statements for data sets representing IMS databases that are not to be dynamically allocated. See “DD Statement Descriptions” on page 303.

DFSESL DD <sup>1</sup>	MATRIXB DD
DFSOLPnn DD	MODBLKSA DD
DFSOLSNnn DD	MODBLKSB DD
DFSWADS <sub>n</sub> DD	MODSTAT DD
IMSACBA DD	PRINTDD DD
IMSACBB DD	PROCLIB DD
IMSIRD DD	RECON1 DD <sup>1</sup>
IMSMON DD	RECON2 DD <sup>1</sup>
IMSRDS DD	RECON3 DD <sup>1</sup>
MATRIXA DD	STEPLIB DD
	SYSUDUMP DD

**Note:** <sup>1</sup> This DD statement is optional.

---

## DBRC Procedure

| The DBRC procedure is an online execution procedure that initiates the DBRC  
| address space.

## DBRC Procedure Description

The procedure shown in Figure 16 on page 204 initializes the DBRC subordinate address space and is executed by IMS via an z/OS START command during control region initialization.

```

//      PROC RGN=64M,DPTY='(14,15)',SOUT=A,
//      IMSID=SYS3,SYS2=
//IEFPROC EXEC PGM=DFSMVRC0,REGION=&RGN,
//      DPRTY=&DPTY,PARM='DRC,&IMSID'
//*****
//*
//STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//PROCLIB DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
//JCLOUT  DD SYSOUT=(A,INTRDR)
//JCLPDS  DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
//SYSUDUMP DD SYSOUT=&SOUT
//SYSABEND DD SYSOUT=&SOUT
//*
//***** DBRC RECON DD CARDS *****
//*
//* USER MAY OPTIONALLY SUPPLY THE DD CARDS
//* REQUIRED FOR THE DBRC RECON DATA SET.
//* IF NO DD STATEMENTS ARE SUPPLIED FOR RECON
//* DATASETS, IMS ASSUMES THAT THE DATASETS
//* HAVE BEEN DESCRIBED THROUGH THE DFSMDA MACRO.

```

Figure 16. Procedure to Initialize the DBRC Subordinate Address Space

IMS system definition stores the cataloged procedure for the DBRC subordinate address space in the IMS.PROCLIB member specified in the DBRCNM=*member* parameter on the IMSCTRL macro. This *member* must be copied to SYS1.PROCLIB.

The IMS control region automatically initiates the DBRC address space procedure. The PDS member name that IMS uses for the DBRC procedure is defined by the following and can be overridden as noted:

1. DBRC is the default name
2. The value specified on the DBRCNM= keyword in the IMSCTRL macro overrides the default name
3. The value specified on the DBRCNM= keyword in the DFSPBxxx PROCLIB member overrides the value specified in the IMSCTRL macro
4. The value specified on the DBRCNM= keyword in JCL EXEC parameter overrides the value specified in the DFSPBxxx PROCLIB member

**Automatic initiation:** An z/OS START command is issued for DBRCNM. The z/OS START command is coded to override the PARM=(DRC,&IMSID) procedure statement. The override defines the IMSID of the subsystem issuing the START command. This allows a generic procedure to be used by multiple IMS control regions.

IMS does not issue the IMS READY message until the DBRC address space successfully initiates. If the DBRC procedure started by the control region is in error, correct the procedure and restart it.

When the DBRC address space connects to the IMS control region, the DBRC procedure name is compared with DBRCNM and if they are not the same, the DBRC address space abends.

**Manual initiation:** The user can supply a START command to be issued instead of the internally defined START. The START command is supplied in member =DBRCNM of a PDS defined in //PROCLIB in the control region JCL. If columns 1-5 are not

“START”, message DFS1930I is issued and the internal START command is issued as if the DBRCNM member was not found in PROCLIB.

**Example:** Following are examples of START commands a user can specify:

- START IMSRDR,MBR=DBRC: the PDS member specified is read in as a job through the z/OS internal reader. It starts as a job rather than as a started task.
- START DBRC,IMSID=IMSA: DBRC executes as a started task. DBRC connects to IMSA.

When the DBRC address space connects to the IMS control region, the DBRC procedure name is compared with the DBRCNM= specification. If they are not the same, the DBRC address space abends.

## DBRC Parameters

The following parameters are valid for the DBRC procedure. See “Parameter Descriptions for IMS Supplied Procedures” on page 255.

DPRTY=	RGN=
IMSID=	SOUT=
IMSPLEX=	SYS2=

## DBRC DD Statements

The following DD statements are valid for the DBRC procedure. See “DD Statement Descriptions” on page 303.

JCLOUT DD	STEPLIB DD
JCLPDS DD	SYSABEND DD
PROCLIB DD	SYSUDUMP DD
RECON <sub>n</sub> DD <sup>1</sup>	

**Note:** <sup>1</sup> This DD statement is optional.

---

## DCC Procedure

The DCC procedure is an online execution procedure to initialize the DCCTL environment.

## DCC Procedure Description

The following procedure executes an IMS DCCTL address space.

```
//      PROC  RGN=64M,SOUT=A,DPTY='(14,15)',
//          SYS=,SYS1=,SYS2=,
//          LOGT=2400,
//          RGSUF=DCC,PARM1=,PARM2=
//IEFPROC EXEC PGM=DFSMVRC0,DPRTY=&DPTY,REGION=&RGN,
//          PARM='DCC,&RGSUF,&PARM1,&PARM2'
//*
//*****
//*                                     *
//* THE DESCRIPTION AND CHARACTERISTICS OF EACH *
//* PARAMETER IS DEFINED AS FOLLOWS: *
//*                                     *
//*****
//*
//***** CONTROL REGION SPECIFICATIONS *****
//*
```

```

|          /** RGSUF   XXX   EXEC PARM DEFAULT BLOCK SUFFIX FOR
|          /**                               MEMBER DFSPBXXX.
|          /*******
|          /**
|          /** PARM1 , PARM2 PARAMETERS BOTH ARE USED TO SPECIFY
|          /** CHARACTER STRINGS THAT CONTAIN IMS KEYWORD
|          /** PARAMETERS. I.E. PARM1='AUTO=Y,PST=222,RES=Y'
|          /**
|          /** ALL OF THE VALID DCCTL KEYWORD PARAMETERS
|          /** ARE DESCRIBED BELOW
|          /*******
|          /** APPLID1 XXXXXXXX VTAM APPLID OF ACTIVE IMS SYSTEM
|          /** APPLID2 XXXXXXXX VTAM APPLID OF XRF ALTERNATE SYSTEM
|          /** APPLID3 XXXXXXXX VTAM APPLID OF RSR TRACKING SYSTEM
|          /** RES      X      BLOCK RESIDENT (N = NO, Y = YES)
|          /** FRE      XXXXX  NUMBER OF FORMAT REQUEST ELEMENTS
|          /** PST      XXX    NUMBER OF PST'S PERMANENTLY ALLOC
|          /** MAXPST  XXX    MAXIMUM NUMBER OF PST'S
|          /** SAV      XXX    NUMBER OF DYNAMIC SAVE AREA SETS
|          /** SRCH    X      IMODULE LOAD LIBRARY SEARCH INDICATOR
|          /**                               0 = SEARCH JPA AND LIBS BEFORE LPA
|          /**                               1 = SEARCH JPA AND LPA BEFORE LIBS
|          /**
|          /** SOD      X      SPIN OFF DUMP SYSOUT CLASS
|          /** VAUT    X      VTAM AUTH PATH OPTION (1=YES,0=NO)
|          /** FMTO
|          /**          T = ONLINE FORMATTED DUMP WITH
|          /**          STORAGE IMAGE DELETIONS.
|          /**          OFFLINE SDUMPS PERMITTED FOR
|          /**          NON-IMS TERMINATING ERRORS.
|          /**          P = FULL ONLINE FORMATTED DUMP.
|          /**          OFFLINE SDUMPS PERMITTED FOR
|          /**          NON-IMS TERMINATING ERRORS.
|          /**          F = FULL ONLINE FORMATTED DUMP.
|          /**          OFFLINE SDUMPS SUPPRESSED FOR
|          /**          NON-IMS TERMINATING ERRORS.
|          /**          N = NO FORMATTED DUMP, NO OFFLINE
|          /**          DUMP. OFFLINE SDUMPS PERMITTED
|          /**          FOR NON-IMS TERMINATING ERRORS
|          /**          Z = NO FORMATTED DUMP, NO OFFLINE
|          /**          DUMP. OFFLINE SDUMPS
|          /**          SUPPRESSED FOR NON-IMS
|          /**          TERMINATING ERRORS.
|          /** (DEFAULT) D = OFFLINE DUMP, OR ONLINE FOR-
|          /**          MATTED DUMP WITH STORAGE IMAGE
|          /**          DELETIONS IF OFFLINE DUMPING
|          /**          FAILS. OFFLINE SDUMPS
|          /**          PERMITTED FOR NON-IMS
|          /**          TERMINATING ERRORS.
|          /**          X = OFFLINE DUMP, OR ONLINE FOR-
|          /**          MATTED DUMP WITH STORAGE IMAGE
|          /**          DELETIONS IF OFFLINE DUMPING
|          /**          FAILS. OFFLINE SDUMPS
|          /**          SUPPRESSED FOR NON-IMS
|          /**          TERMINATING ERRORS.
|          /**          M = OFFLINE DUMP, ONLINE IMS DUMP
|          /**          FORMATTING NOT PERMITTED.
|          /**          OFFLINE SDUMPS PERMITTED FOR
|          /**          NON-IMS TERMINATING ERRORS.
|          /**          R = OFFLINE DUMP, ONLINE IMS DUMP
|          /**          FORMATTING NOT PERMITTED.
|          /**          OFFLINE SDUMPS SUPPRESSED FOR
|          /**          NON-IMS TERMINATING ERRORS.

```



```

|          /** AUTO X AUTOMATIC RESTART OPTION
|          /** Y = AUTOMATIC RESTART DESIRED
|          /** N = NO AUTOMATIC RESTART
|          /** IMSID XXXX IMS DCCTL SUBSYSTEM ID
|          /** NLXB XXX NUMBER OF PARALLEL VTAM SESSIONS
|          /** APPC X Y = ACTIVATE APPC/IMS
|          /** N = DO NOT ACTIVATE APPC/IMS
|          /** LTERM X Y = LTERM USED IN DFSAPPC PROCESS
|          /** N = LTERM NOT USED IN DFSAPPC
|          /** PROCESS
|          /** ARMRST X Y = ALLOW MVS ARM TO RESTART
|          /** N = ARM NOT RESTART IMS
|          /** RRS X Y = ENABLE PROT CONV SUPPORT
|          /** N = DISABLE PROT CONV SUPPORT
|          /** DBRCNM XXXXXXXX DBRC PROCLIB MEMBER NAME
|          /** PRDR XXXXXXXX IMSRDR PROCLIB MEMBER NAME
|          /** SSM XXXX EXT SUBSYSTEM PROCLIB MEMBER ID
|          /** WADS X SINGLE OR DUAL WRITE AHEAD OPTION
|          /** S=SINGLE, D=DUAL
|          /** ARC XX AUTOMATIC ARCHIVING OPTION
|          /** 0 = NOT AUTOMATIC
|          /** 1-99 = AUTOMATIC
|          /** RECA XXX RECEIVE ANY BUFFERS
|          /** FESTIM XXXX FRONTENDSWITCH TIMEOUT
|          /** RECASZ XXXXX RECEIVE ANY BUFFER SIZE
|          /** CRC X COMMAND RECOGNITION CHARACTER
|          /** TSR X U = UTC TIME
|          /** L = LOCAL TIME (DEFAULT)
|          /** YEAR4 X N = 2-DIGIT DATE (DEFAULT)
|          /** Y = 4-DIGIT DATE
|          /** DC XXX DC PROC MEMBER SUFFIX IN
|          /** IMS.PROCLIB
|          /** DEFAULT VALUE IS 000
|          /** CPLOG XXXXXX CHECKPOINT LOG INTERVAL
|          /** OR
|          /** CPLOG XXM CHECKPOINT LOG INTERVAL
|          /** PASSWD1 XXXXXXXX VTAM ACB PASSWORD
|          /** ORSMBR XX SUFFIX FOR ORS MEMBER
|          /** IMSGROUP XXXX 4 CHAR USER SPEC NAME
|          /** DESC XX MSG DESC CODE
|          /** MCS (XX,XX) MSG ROUTE CODES
|          /** SVC2 XXX TYPE 2 SVC NUMBER
|          /******* OTMA PARAMETERS *****
|          /**
|          /** OTMA X Y = OTMA ENABLED
|          /** N = OTMA NOT ENABLED
|          /** DEFAULT VALUE IS N
|          /** OTMANM XXXXXXXX IMS OTMA XCF MEMBER NAME
|          /** GRNAME XXXXXXXX OTMA XCF GROUP NAME
|          /** NO DEFAULT VALUE
|          /** GRSNAME XXXXXXXX GENERIC RESOURCE GROUP
|          /** NAME
|          /** NO DEFAULT VALUE
|          /**
|          /******* SECURITY PARAMETERS *****
|          /**
|          /** AOIS X ICMD SECURITY OPTION
|          /** AOI1 X CMD SECURITY OPTION
|          /** A = ALL
|          /** N = NONE
|          /** C = DFSCCMD0 EXIT

```

```

|          /**          R = RACF
|          /**          S = SMU
|          /**   TCORACF X   TCO RACF SECURITY OPTION
|          /**          Y = YES
|          /**          N = NO
|          /**   APPCSE  X   C = APPC RACF SECURITY IS CHECK
|          /**          F = APPC RACF SECURITY IS FULL
|          /**          N = APPC RACF SECURITY IS NONE
|          /**          P = APPC RACF SECURITY IS PROFILE
|          /**   CMDMCS  X   MCS/EMCS COMMAND OPTION
|          /**          N=COMMANDS NOT ALLOWED WITH CRC
|          /**          Y=ALL COMMANDS ALLOWED WITH CRC
|          /**          R=RACF COMMAND SECURITY
|          /**          C=DFSCCMD0 COMMAND SECURITY
|          /**          B=RACF AND DFSCCMD0 CMD SEC
|          /**   ISIS    X   RAS AND AGN SECURITY OPTION
|          /**          N = NO RESOURCE ACCESS SECURITY
|          /**          R = RACF RESOURCE ACCESS SECURITY
|          /**          C = RACF RESOURCE ACCESS SECURITY
|          /**          A = RACF RESOURCE ACCESS SECURITY
|          /**          0 = NO AGN SECURITY
|          /**          1= APPLICATION GROUP NAME SECURITY
|          /**          2= APPLICATION GROUP NAME SECURITY
|          /**   RCF     X   RACF TRAN/CMD AUTHORIZATION
|          /**          N = NO TRAN/CMD AUTHORIZATION
|          /**          C = ETO CMD AUTHORIZATION
|          /**          T = TRAN AUTHORIZATION
|          /**          S = STATIC LTERM CMD AUTH
|          /**          Y = C AND T
|          /**          A = C AND T AND S
|          /**          B = 'A' + SVST TABLE NOT LOADED
|          /**          R = 'S' + SVST TABLE NOT LOADED
|          /**   RVFY    X   RACF REVERIFY OPTION
|          /**          Y = YES, N = NO
|          /**   SGN     X   SIGNON AUTHORIZATION CHECKING
|          /**          F = MTO CANNOT NEGATE LOADING
|          /**          SIGNON VERIFICATION SECURITY
|          /**          TABLE(SVST) FROM IMS.MATRIX
|          /**          Y = SVST LOADED UNLESS OVERRIDEN
|          /**          BY MTO
|          /**          N = SVST NOT LOADED UNLESS
|          /**          OVERRIDDEN BY MTO
|          /**          M = SINGLE USERID CAN SIGNON
|          /**          TO MULTIPLE STATIC TERMINALS
|          /**          G = 'F' + 'M'
|          /**          Z = 'Y' + 'M'
|          /**          D = 'F' + SVST TABLE NOT LOADED
|          /**          E = 'G' + SVST TABLE NOT LOADED
|          /**          W = 'Y' + SVST TABLE NOT LOADED
|          /**          X = 'Z' + SVST TABLE NOT LOADED
|          /**   TRN     X   TRANSACTION AUTHORIZATION CHECKING
|          /**          F = FORCED, Y = YES, N = NO
|          /**          E = 'F' + SVST TABLE NOT LOADED
|          /**          X = 'Y' + SVST TABLE NOT LOADED
|          /**   RCFTCB  XX  NUMBER OF RCF TCB'S
|          /**
|          /******* MESSAGE QUEUE PARAMETERS *****
|          /**
|          /**   EXVR    X   PAGEFIX QMGR BUFFER POOLS
|          /**          (1=YES, 0=NO)
|          /**   QBUF    XXXX NUMBER OF MESSAGE QUEUE BUFFERS

```

```

|          /** QTU   XXX   MSG QUEUE UPPER THRESHOLD (%)
|          /** QTL   XXX   MSG QUEUE LOWER THRESHOLD (%)
|          /**
|          /******* SHARED QUEUES PARAMETERS *****
|          /**
|          /** LGMSGZ XXXXX LONG MESSAGE SIZE
|          /** QBUFHITH XXX MSG QBUF HIGH THESHOLD %
|          /** QBUFLWTH XXX MSG QBUF LOW THESHOLD %
|          /** QBUFMAX XXXX MAX NUMBER OF MSG QUEUE BUFFERS
|          /** QBUFPCTX XXX  % MSG QBUF DYNAMIC EXPAND
|          /**                               WHEN QBUFHITH EXCEEDED
|          /**                               DEFAULT IS 20%
|          /** QBUFSZ XXXXX SIZE OF MESSAGE QUEUE BUFFERS
|          /** SHMSGZ XXXXX SHORT MESSAGE SIZE
|          /** SHAREDQ XXX  SQ PROC MEMBER SUFFIX IN
|          /**                               IMS.PROCLIB
|          /**                               NO DEFAULT VALUE
|          /**
|          /******* ETO PARAMETERS *****
|          /**
|          /** ETO   X    Y = EXTENDED TERMINAL OPTION
|          /**                               N = NO EXTENDED TERMINAL OPTION
|          /**                               M = NO EXTENDED TERMINAL OPTION
|          /**                               BUT LOGON USERDATA SUPPORTED
|          /**                               FOR STATIC TERMINALS
|          /** ASOT   XXXX ETO AUTO SIGNOFF TIME
|          /** ALOT   XXXX ETO AUTO LOGNOFF TIME
|          /** DLQT   XXX  ETO DEAD LETTER QUEUE SIZE
|          /**
|          /******* RSR PARAMETERS *****
|          /**
|          /** RSRMBR XX   SUFFIX FOR RSR MEMBER
|          /** TRACK  XXX  NO = NO RECOVERY TRACKING DONE
|          /**                               RLT = RECOVERY TRACKING DONE
|          /** USERVAR XXXXXXXX USER NAME OF ACTIVE IMS SYSTEM FOR RSR
|          /**
|          /******* HASH TABLE PARAMETERS *****
|          /**
|          /** LHTS   XXXXX # OF CNT HASH TABLE SLOTS
|          /** NHTS   XXXXX # OF VTCB HASH TABLE SLOTS
|          /** UHTS   XXXXX # OF SPQB HASH TABLE SLOTS
|          /**
|          /******* FAST PATH PARAMETERS *****
|          /**
|          /** EPCB   XXXX  EPCB POOL SIZE (1K BLOCKS)
|          /** EMHL   XXXXX SIZE OF EMH BUFFER IN BYTES
|          /**
|          /******* STORAGE POOL VALUES IN K, M OR G *****
|          /**
|          /** FBP    XXXXXX MESSAGE FORMAT BLOCK POOL SIZE
|          /** PSB    XXXXXX DCCTL PSB POOL SIZE
|          /** CIOP   XXXXXX TP DEVICE I/O POOL UPPER LIMIT
|          /** WKAP   XXXXXX WORKING STORAGE POOL SIZE
|          /** PSBW   XXXXXX PSB WORK POOL SIZE
|          /** HIOP   XXXXXX HIGH I/O POOL UPPER LIMIT
|          /** EMHB   XXXXXX EMHB POOL UPPER LIMIT
|          /** LUMP   XXXXXX LUMP POOL UPPER LIMIT
|          /** LUMC   XXXXXX LUMC POOL UPPER LIMIT
|          /** AOIP   XXXXXX AOI POOL UPPER LIMIT
|          /**
|          /******* MEMBER SUFFIXES *****

```

```

|      /**
|      /**  SUF      X      CONTROL PGM NAME (DCCTL NUCLEUS)
|      /**  FIX      XX     PAGE FIX NAME (DFSFIX..)
|      /**  PRLD     XX     MODULE PRELOAD NAME (DFSAMPL..)
|      /**  VSPEC    XX     TRACE/LOG DEF (DFSISM..)
|      /**  SPM      XX     STORAGE POOL OPTIONS (DFSMPM..)
|      /**  CSLG     XXX    CSL GLOBAL MEMBER (DFSCGXXX)
|      /**  DSCT     X      ETO USER DESCRIPTOR TABLE(DFSDSCTX)
|      /**
|      /******* XRF PARAMETERS *****
|      /**
|      /**  HSBID    X      XRF SYSTEM IDENTIFICATION
|      /**                               1 FOR FIRST SYSTEM
|      /**                               2 FOR SECOND SYSTEM
|      /**  HSBMBR   XX     XRF OPTIONS (DFSHSB..)
|      /**
|      /**  MNPS     XXXXXXXX NAME OF MNPS ACB
|      /**                               USERVAR WILL BE IGNORED
|      /**  MNPSPW   XXXXXXXX MNPS ACB PASSWORD
|      /**
|      /*******
|      /**                               *
|      /**          XRF SYSTEM DATA SET INFORMATION          *
|      /**                               *
|      /*******
|      /**
|      /** THE FOLLOWING DATA SETS MUST RESIDE ON SHARED
|      /** VOLUMES:
|      /**
|      /**      DDNAMES:   DFSOLPXX, DFSOLSXX
|      /**                  DFSWADSX
|      /**                  IMSRDS   - FIRST RDS
|      /**                  IMSRDS2  - SECOND RDS (MANDATORY)
|      /**                  MODSTAT   - FIRST MODSTAT
|      /**                  MODSTAT2  - 2ND MODSTAT (MANDATORY)
|      /**                  RECON1, RECON2, RECON3
|      /**
|      /** THE FOLLOWING DATA SETS MUST BE REPLICATED AND
|      /** SHOULD BE ON NON-SHARED VOLUMES.  IF YOUR XRF
|      /** CONFIGURATION REQUIRES THAT BOTH IMS SUBSYSTEMS
|      /** BE EXECUTABLE ON EITHER CEC, THEN THESE DATA
|      /** SETS MUST BE ON SHARED VOLUMES.
|      /**
|      /**      DDNAMES:   IMSSPA
|      /**                  LGMSG, LGMSG
|      /**                  QBLKS, QBLKSL
|      /**                  SHMSG, SHMSG
|      /**
|      /**                  SYSABEND, SYSUDUMP
|      /**                  IMSMON
|      /**
|      /** THE FOLLOWING DATA SETS MAY BE REPLICATED.  IF
|      /** REPLICATED, THESE DATA SETS SHOULD BE ON
|      /** NON-SHARED VOLUMES.  IF THESE DATA SETS ARE NOT
|      /** REPLICATED, THEY MUST BE ON SHARED VOLUMES.  IF
|      /** YOUR XRF CONFIGURATION REQUIRES THAT BOTH IMS
|      /** SUBSYSTEMS BE EXECUTABLE ON EITHER CEC,
|      /** THESE DATA SETS MUST BE ON SHARED VOLUMES.
|      /**
|      /**      DDNAMES:   FORMATA, FORMATB
|      /**                  IMSACBA, IMSACBB

```

```

|          /*          IMSTFMTA, IMSTFMTB
|          /*          MATRIXA, MATRIXB
|          /*          MODBLKSA, MODBLKSB
|          /*          PROCLIB, RESLIB
|          /*          OTHER STEPLIB DATA SETS
|          /*
|          /* THE FOLLOWING DATA SETS SHOULD ALSO BE CONSIDERED
|          /* FOR REPLICATION ON EITHER SHARED OR NON-SHARED
|          /* VOLUMES (AS APPROPRIATE).
|          /*
|          /*      DATA SET:  JOBS   USED IN THE IMSRDR PROCEDURE
|          /*                  PGMLIB
|          /*                  PSBLIB USED BY GSAM
|          /*                  DBDLIB USED BY GSAM
|          /*
|          /* THE FOLLOWING IMS DATA SETS MAY ALSO BE IMPACTED
|          /* BY THE CONFIGURATION OF YOUR XRF COMPLEX.
|          /*
|          /* INSTALIB - USED DURING INSTALLATION/IVP
|          /* RESLIB  - CREATED BY SYSDEF - MANAGED BY SMP
|          /* PROCLIB - CREATED BY SYSDEF
|          /* MACLIB  - CREATED BY SYSDEF - MANAGED BY SMP
|          /* OPTIONS - CREATED BY SYSDEF - USED BY SMP
|          /* OBJDSET - CREATED BY SYSDEF
|          /* MODBLKS - CREATED BY SYSDEF - MANAGED BY SMP
|          /* MATRIX  - ONLINE CHANGE STAGING LIBRARY
|          /* ACBLIB  - ONLINE CHANGE STAGING LIBRARY
|          /* FORMAT  - ONLINE CHANGE STAGING LIBRARY
|          /* TFORMAT - ONLINE CHANGE STAGING LIBRARY
|          /* REFERAL - USED IN CONJUNCTION WITH FORMAT
|          /* DBSOURCE - USED BY SYSDEF - MANAGED BY SMP
|          /* DCSOURCE - USED BY SYSDEF - MANAGED BY SMP
|          /* SRSOURCE - USED BY SYSDEF - MANAGED BY SMP
|          /* ADFSMAC  - USED BY SYSDEF - MANAGED BY SMP
|          /* LOAD    - USED BY SYSDEF - MANAGED BY SMP
|          /* JCLLIB  - MANAGED BY SMP
|          /* SMP DATA SETS
|          /*
|          /******
|          /*          *
|          /*      DATA DEFINITION STATEMENTS FOLLOW          *
|          /*          *
|          /******
|          /*
|          /****** LIBRARY STATEMENTS *****
|          /*
|          /*STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
|          /*PROCLIB DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
|          /*
|          /******
|          /*
|          /* IN ORDER TO START A DEPENDENT REGION, MODIFIED
|          /* START-UP JCL IS WRITTEN FROM INTERNAL STORAGE TO
|          /* THE INTERNAL READER.
|          /*
|          /*IMSIRD  DD SYSOUT=(A,INTRDR)
|          /*
|          /****** DASD LOGGING STATEMENTS *****
|          /*
|          /* THE FOLLOWING DD CARDS DESCRIBE THE DASD LOGGING
|          /* OLDS AND WADS. THESE CARDS ARE FOR EXAMPLE ONLY.

```

```

|      /** ALL OLDS AND WADS DATA SETS MAY BE DYNAMICALLY
|      /** ALLOCATED. DD CARDS ARE NOT REQUIRED.
|      /** THE OLDS AND WADS TO BE USED DURING STARTUP MUST
|      /** BE SPECIFIED VIA OLDSDEF AND WADSDEF CONTROL
|      /** STATEMENTS IN THE DFSVSMXX MEMBER OF IMS PROCLIB.
|      /** THE ACTUAL SELECTION OF OLDS AND WADS MUST BE
|      /** TAILORED TO INSTALLATION REQUIREMENTS. THE OLDS
|      /** AND WADS MUST BE PREDEFINED BY A SET UP JOB.
|      /** THE BLOCK SIZE OF ALL OLDS MUST BE THE SAME.
|      /** THE BLOCK SIZE AND DEVICE TYPE OF ALL WADS MUST
|      /** BE THE SAME. AT LEAST 3 PRIMARY OLDS AND 1 WADS
|      /** MUST BE AVAILABLE FOR STARTUP. THE BLOCK SIZE
|      /** SHOULD NOT BE SPECIFIED IN THIS JCL. THE LOGGER
|      /** WILL GET THE BLOCK SIZE FROM THE VTOC.
|      /**
|      /**DFSOLP00 DD DSN=IMS.&SYS.OLP00,DISP=SHR
|      /**DFSOLP01 DD DSN=IMS.&SYS.OLP01,DISP=SHR
|      /**DFSOLP02 DD DSN=IMS.&SYS.OLP02,DISP=SHR
|      /**DFSOLP03 DD DSN=IMS.&SYS.OLP03,DISP=SHR
|      /**DFSOLP04 DD DSN=IMS.&SYS.OLP04,DISP=SHR
|      /**DFSOLP05 DD DSN=IMS.&SYS.OLP05,DISP=SHR
|      /**
|      /**DFSOLS00 DD DSN=IMS.&SYS.OLS00,DISP=SHR
|      /**DFSOLS01 DD DSN=IMS.&SYS.OLS01,DISP=SHR
|      /**DFSOLS02 DD DSN=IMS.&SYS.OLS02,DISP=SHR
|      /**DFSOLS03 DD DSN=IMS.&SYS.OLS03,DISP=SHR
|      /**DFSOLS04 DD DSN=IMS.&SYS.OLS04,DISP=SHR
|      /**DFSOLS05 DD DSN=IMS.&SYS.OLS05,DISP=SHR
|      /**
|      /**DFSWADS0 DD DSN=IMS.&SYS.WADS0,DISP=SHR
|      /**DFSWADS1 DD DSN=IMS.&SYS.WADS1,DISP=SHR
|      /**
|      /******* MONITOR LOGGING STATEMENTS *****
|      /**
|      /** THE IMSMON DD STATEMENT MUST BE REMOVED IF
|      /** THIS DATA SET IS TO BE DYNAMICALLY ALLOCATED.
|      /**
|      /**IMSMON DD DSN=IMS.&SYS1.IMSMON,DISP=(,KEEP),
|      /** VOL=(,,99),UNIT=(&LOGT,,DEFER)
|      /**
|      /******* MESSAGE QUEUE STATEMENTS *****
|      /**
|      /**QBLKS DD DSN=IMS.&SYS1.QBLKS,DISP=OLD
|      /**SHMSG DD DSN=IMS.&SYS1.SHMSG,DISP=OLD
|      /**LGMSG DD DSN=IMS.&SYS1.LGMSG,DISP=OLD
|      /**QBLKSL DD DSN=IMS.&SYS1.QBLKSL,DISP=OLD
|      /**SHMSGSL DD DSN=IMS.&SYS1.SHMSGSL,DISP=OLD
|      /**LGMSGSL DD DSN=IMS.&SYS1.LGMSGSL,DISP=OLD
|      /**
|      /******* ONLINE CHANGE STATEMENTS *****
|      /**
|      /**IMSACBA DD DSN=IMS.&SYS2.ACBLIBA,DISP=SHR
|      /**IMSACBB DD DSN=IMS.&SYS2.ACBLIBB,DISP=SHR
|      /**MODBLKSA DD DSN=IMS.&SYS2.MODBLKSA,DISP=SHR
|      /**MODBLKSB DD DSN=IMS.&SYS2.MODBLKSB,DISP=SHR
|      /**MODSTAT DD DSN=IMS.&SYS.MODSTAT,DISP=SHR
|      /**MODSTAT2 DD DSN=IMS.&SYS.MODSTAT2,DISP=SHR
|      /**
|      /******* MFS STATEMENTS *****
|      /**
|      /**FORMATA DD DSN=IMS.&SYS2.FORMAT,DISP=SHR

```

```

| //FORMATB DD DSN=IMS.&SYS2.FORMATB,DISP=SHR
| //IMSTFMTA DD DSN=IMS.&SYS2.TFORMAT,DISP=SHR
| // DD DSN=IMS.&SYS2.FORMATA,DISP=SHR
| //IMSTFMTB DD DSN=IMS.&SYS2.TFORMAT,DISP=SHR
| // DD DSN=IMS.&SYS2.FORMATB,DISP=SHR
| //*
| //***** DCCTL SYSTEM STATEMENTS *****
| //*
| //IMSRDS DD DSN=IMS.&SYS.RDS,DISP=SHR
| //IMSRDS2 DD DSN=IMS.&SYS.RDS2,DISP=SHR
| //MATRIXA DD DSN=IMS.&SYS2.MATRIXA,DISP=SHR
| //MATRIXB DD DSN=IMS.&SYS2.MATRIXB,DISP=SHR
| // *DFSTCF DD DSN=IMS.TCFSLIB,DISP=SHR
| //PRINTDD DD SYSOUT=&SOUT
| //SYSUDUMP DD SYSOUT=&SOUT,
| // DCB=(LRECL=125,RECFM=FBA,BLKSIZE=3129),
| // SPACE=(6050,300,,ROUND)
| //*
| //***** TELEPROCESSING LINE STATEMENTS *****
| //***** GENERATED FROM DCCTL DEFINITION *****
| //*
| // *** THE GENERATED TELEPROCESSING LINE STATEMENTS WOULD BE HERE ***

```

## DCC Parameters

The following parameters are valid for the DCC procedure. See “Parameter Descriptions for IMS Supplied Procedures” on page 255.

ALOT=	GRSNAME=	RCFTCB=
AOIP=	HIOP=	RECA=
AOIS=	HSBID=	RECASZ=
I AOI1=	HSBMBR=	RES=
APPC=	IMSGROUP=	RGN=
APPCSE=	IMSID=	RGSUF=
APPLID1=	ISIS=	RRS=
APPLID2=	LGMSGSZ=	RSRMBR=
APPLID3=	LHTS=	RVFY=
ARC=	LTERM=	SAV=
ARMRST=	LUMC=	SGN=
ASOT=	LUMP=	SHAREQ=
AUTO=	MAXPST=	SHMSGSZ=
I CIOP=	MCS=	SOD=
CMDMCS=	NHTS=	SOUT=
CPLOG=	NLXB=	SPM=
CRC=	OTMA=	SRCH=
CSLG=	OTMAASY=	SSM=
DBRCNM=	OTMAMD=	SUF=
I DC=	OTMANM=	SVC2=
I DESC=	OTMASP=	SYS=
DSCT=	PASSWD1=	SYS1=
EMHB=	PRDR=	SYS2=
I EMHL=	PRLD=	TCORACF=
EPCB=	PSB=	TRACK=
ETO=	PSBW=	TRN=
EXVR=	PST=	TSR=
FBP=	QBUF=	UHTS=
FESTIM=	QBUFHITH=	USERVAR=
FIX=	QBUFLWTH=	VAUT=
FMTO=	QBUFMAX=	VSPEC=
FRE=	QBUFPCX=	WADS=
GRNAME=	QBUFSZ=	WKAP=
	QTL=	YEAR4=
	QTU=	
	RCF=	

## DCC DD Statements

The following DD statements are valid for the DCC procedure. See "DD Statement Descriptions" on page 303.



DFSDUPPM DD	MODBLKSA DD
DFSESL DD <sup>1</sup>	MODBLKSB DD
DFSOLP <sub>n</sub> DD	MODSTAT DD
DFSOLS <sub>n</sub> DD	MODSTAT2 DD
DFSTCF DD <sup>1</sup>	PRINT DD
FORMATA DD	PROCLIB DD
FORMATB DD	QBLKS DD
IMSACBA DD	QBLKSL DD
IMSACBB DD	RECON1 DD <sup>1</sup>
IMSIRD DD	RECON2 DD <sup>1</sup>
IMSMON DD	RECON3 DD <sup>1</sup>
IMSRDS DD	SHMSG DD
IMSRDS2 DD	SHMSGL DD
IMSTFMTA DD	STEPLIB DD
IMSTFMTB DD	SYSABEND DD
LGMSG DD	SYSUDUMP DD
LGMSGL DD	
MATRIXA DD	
MATRIXB DD	

**Note:** <sup>1</sup> These DD statements are optional.

---

## DFSJBP Procedure

The DFSJBP procedure starts a Java non-message driven dependent region (hereafter called a JBP region) that resembles a non-message driven BMP region (for example, similar procedure parameters and OS/390 TCB structure) with the following exceptions:

- Only a Java application can be scheduled in this region.
- The JVMOPMAS=<*member name*> procedure parameter, which sets the name of the member in IMS.PROCLIB that contains the JVM options for the standalone JVM, is specific to initializing a JVM. JVMOPMAS is a required parameter.
- The ENVIRON= parameter that specifies the name of the PROCLIB member that contains the environment settings. ENVIRON= is a required parameter.
- The existing IN=, PRLD=, and SSM= parameters on the IMSBATCH procedure are not supported on the DFSJBP procedure.

The DFSJBP procedure resides in IMS.PROCLIB as a standard IMS System Definition-supplied procedure and can be invoked as IMSBATCH would be invoked.

The DFSJVMAP PROCLIB member can be used in conjunction with a DFSJBP procedure. DFSJVMAP maps all the 8-byte or less uppercase IMS Java application names (specified to IMS) to the true OMVS path name for the “.class” file associated with that IMS Java application. For more information, see “Member DFSJVMAP” on page 362.

## DFSJBP Procedure Description

The procedure shown in Figure 17 on page 216 starts a JBP region.

```

|          //          PROC MBR=TEMPNAME,PSB=,JVMOPMAS=,OUT=,
|          //          OPT=N,SPIE=0,TEST=0,DIRCA=000,
|          //          STIMER=,CKPTID=,PARDLI=,
|          //          CPUTIME=,NBA=,OBA=,IMSID=,AGN=,
|          //          PREINIT=,RGN=56K,SOUT=A,
|          //          SYS2=,ALTID=,APARM=,ENVIRON=,LOCKMAX=,
|          //          XPLINK=N
|          // *
|          //JBPRGN EXEC PGM=DFSRR00,REGION=&RGN,
|          //          PARM=(JBP,&MBR,&PSB,&JVMOPMAS,&OUT,
|          //          &OPT&SPIE&TEST&DIRCA,
|          //          &STIMER,&CKPTID,&PARDLI,&CPUTIME,
|          //          &NBA,&OBA,&IMSID,&AGN,
|          //          &PREINIT,&ALTID,
|          //          '&APARM',&ENVIRON,&LOCKMAX,
|          //          &XPLINK)
|          //STEPLIB DD DSN=IMS.&SYS2.SDFSJLIB,DISP=SHR
|          //          DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
|          //          DD DSN=IMS.&SYS2.PGMLIB,DISP=SHR
|          //          DD DSN=CEE.SCEERUN,DISP=SHR
|          //          DD DSN=SYS1.CSSLIB,DISP=SHR
|          //PROCLIB DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
|          //SYSUDUMP DD SYSOUT=&SOUT,
|          //          DCB=(LRECL=121,RECFM=VBA,BLKSIZE=3129),
|          //          SPACE=(125,(2500,100),RLSE,,ROUND)

```

Figure 17. Procedure to Start a JBP Region

## DFSJBP Parameters

The following parameters are valid for the DFSJBP procedure. See “Parameter Descriptions for IMS Supplied Procedures” on page 255.

AGN=	OBA=
ALTID=	OPT=
APARM=	OUT=
CKPTID=	PARDLI=
CPUTIME=	PREINIT=
DIRCA=	PSB=
ENVIRON=	RGN=
IMSID=	SOUT=
JVMOPMAS=	SPIE=
LOCKMAX=	SYS2=
MBR=	TEST=
NBA=	XPLINK=

## DFSJBP DD Statements

The following DD statements are valid for the DFSJBP procedure. See “DD Statement Descriptions” on page 303.

DFSDB2AF DD	PROCLIB DD
JAVAERR DD	STEPLIB DD
JAVAIN DD	SYSUDUMP DD
JAVAOUT DD	

---

## DFSJMP Procedure

The DFSJMP procedure starts a Java message-driven dependent region (hereafter called a JMP region) that resembles an MPP region (for example, similar procedure parameters and OS/390 TCB structure) with the following exceptions:

- The region can only schedule a Java application.
- The JVMOPWKR=<*member name*> sets the name of the member in IMS.PROCLIB that contains the JVM options for the worker JVM. JVMOPWKR is an optional parameter.
- The JVMOPMAS=<*member name*> sets the name of the member in IMS.PROCLIB that contains the JVM options for the master JVM. JVMOPMAS is a required parameter.
- The ENVIRON= parameter that specifies the name of the PROCLIB member that contains the environment settings. ENVIRON= is a required parameter.
- The existing APPLFE=, DBLDL=, PRLD=, SSM=, VSFX=, and VFREE= parameters on the DFSMPR procedure are not supported on the DFSJMP procedure.

The DFSJMP procedure resides in IMS.PROCLIB as a standard IMS System Definition-supplied procedure and can be invoked as DFSMPR would be invoked.

The DFSJVMAP PROCLIB member can be used in conjunction with a DFSJMP procedure. DFSJVMAP maps all the 8-byte or less uppercase IMS Java application names (specified to IMS) to the true OMVS path name for the “.class” file associated with that IMS Java application. For more information, see “Member DFSJVMAP” on page 362.

## DFSJMP Procedure Description

The procedure shown in Figure 18 on page 218 starts a JMP region.

```

|           //      PROC SOUT=A,RGN=56K,SYS2=,
|           //      CL1=001,CL2=000,CL3=000,CL4=000,
|           //      OPT=N,OVLA=0,SPIE=0,VALCK=0,TLIM=00,
|           //      PCB=000,STIMER=,SOD=,
|           //      NBA=,OBA=,IMSID=,AGN=,
|           //      PREINIT=,ALTID=,PWFI=N,APARM=,
|           //      LOCKMAX=,ENVIRON=,JVMOPWKR=,
|           //      JVMOPMAS=,XPLINK=N
|           /**
|           //JMPRGN EXEC PGM=DFSRR00,REGION=&RGN,
|           //      TIME=1440,DPRTY=(12,0),
|           //      PARM=(JMP,&CL1&CL2&CL3&CL4,
|           //      &OPT&OVLA&SPIE&VALCK&TLIM&PCB,
|           //      &STIMER,&SOD,&NBA,
|           //      &OBA,&IMSID,&AGN,&PREINIT,
|           //      &ALTID,&PWFI,'&APARM',&LOCKMAX,
|           //      &ENVIRON,&JVMOPWKR,&JVMOPMAS,
|           //      &XPLINK)
|           /**
|           //STEPLIB DD DSN=IMS.&SYS2.PGMLIB,DISP=SHR
|           //      DD DSN=IMS.&SYS2.SDFSJLIB,DISP=SHR
|           //      DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
|           //      DD DSN=CEE.SCEERUN,DISP=SHR
|           //      DD DSN=SYS1.CSSLIB,DISP=SHR
|           //PROCLIB DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
|           //SYSUDUMP DD SYSOUT=&SOUT,
|           //      DCB=(LRECL=121,BLKSIZE=3129,RECFM=VBA),
|           //      SPACE=(125,(2500,100),RLSE,,ROUND)

```

Figure 18. Procedure to Start a JMP Region

## DFSJMP Parameters

The following parameters are valid for the DFSJMP procedure. See “Parameter Descriptions for IMS Supplied Procedures” on page 255.

AGN=	OVLA=
ALTID=	PCB=
APARM=	PREINIT=
CL1=,CL2=,CL3=,CL4=	PWFI=
ENVIRON=	RGN=
IMSID=	SOD=
JVMOPMAS=	SOUT=
JVMOPWKR=	SPIE=
LOCKMAX=	STIMER=
NBA=	SYS2=
OBA=	TLIM=
OPT=	VALCK=
	XPLINK=

## DFSJMP DD Statements

The following DD statements are valid for the DFSJMP procedure. See “DD Statement Descriptions” on page 303.

DFSD2AF DD	PROCLIB DD
JAVAERR DD	STEPLIB DD
JAVAIN DD	SYSUDUMP DD
JAVAOUT DD	

---

## DFSMPR Procedure

The DFSMPR procedure is an online execution procedure that initiates an IMS message processing address space.

## DFSMPR Procedure Description

The procedure shown in Figure 19 executes an IMS message processing address space.

```

//      PROC SOUT=A,RGN=56K,SYS2=,
//      CL1=001,CL2=000,CL3=000,CL4=000,
//      OPT=N,OVLA=0,SPIE=0,VALCK=0,TLIM=00,
//      PCB=000,PRLD=,STIMER=,SOD=,DBLDL=,
//      NBA=,OBA=,IMSID=,AGN=,VSFX=,VFREE=,
//      SSM=,PREINIT=,ALTID=,PWFI=N,
//      APARM=,LOCKMAX=,APPLFE=
// *
//REGION EXEC PGM=DFSRR00,REGION=&RGN,
//      TIME=1440,DPRTY=(12,0),
//      PARM=(MSG,&CL1&CL2&CL3&CL4,
//      &OPT&OVLA&SPIE&VALCK&TLIM&PCB,
//      &PRLD,&STIMER,&SOD,&DBLDL,&NBA,
//      &OBA,&IMSID,&AGN,&VSFX,&VFREE,
//      &SSM,&PREINIT,&ALTID,&PWFI,
//      '&APARM',&LOCKMAX,&APPLFE)
// *
//STEPLIB DD DSN=IMS.&SYS2.PGMLIB,DISP=SHR
//      DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//PROCLIB DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
//SYSUDUMP DD SYSOUT=&SOUT,
//      DCB=(LRECL=121,BLKSIZE=3129,RECFM=VBA),
//      SPACE=(125,(2500,100),RLSE,,ROUND)

```

Figure 19. Procedure to Execute an IMS Message Processing Address Space

## DFSMPR Parameters

The following parameters are valid for the DFSMPR procedure. See “Parameter Descriptions for IMS Supplied Procedures” on page 255.

AGN=	PRLD=
ALTID=	PWFI=
APARM=	RGN=
APPLFE=	SOD=
CL1=,CL2=,CL3=,CL4=	SOUT=
DBLDL=	SPIE=
IMSID=	SSM=
LOCKMAX=	STIMER=
NBA=	SYS2=
OBA=	TLIM=
OPT=	VALCK=

OVLA=  
PCB=  
PREINIT=

VFREE=  
VSFX=

## DFSMPR DD Statements

The following DD statements are valid for the DFSMPR procedure. See “DD Statement Descriptions” on page 303.

DFSESL DD<sup>1</sup>  
FPTRACE DD<sup>1</sup>  
PROCLIB DD

STEPLIB DD  
SYSUDUMP DD

**Note:** These DD statements are optional.

---

## DLIBATCH Procedure

To run DL/I batch jobs in an RSR environment, you need to be aware of the following:

- The batch-only databases that are identified to be tracked must be included in the system definition (DATABASE macro) for the tracking subsystem.
- The default GSG name and TMI name used by batch jobs are specified in the IMSCTRL macro, but they can be overridden in the DLIBATCH procedure.
- Batch jobs do not use the DFSSRxx PROCLIB member, and are therefore limited to the default VTAM modename (TMDEFLT).

## DLIBATCH Procedure Description

Figure 20 on page 221 shows a one-step procedure for an offline DL/I batch processing program using PSB and DBD libraries.

If VSAM databases are used, refer to “Specifying IMS Buffer Pools” on page 430.

```

|          //          PROC MBR=TEMPNAME,PSB=,BUF=7,
|          //          SPIE=0,TEST=0,EXCPVR=0,RST=0,PRLD=,
|          //          SRCH=0,CKPTID=,MON=N,LOGA=0,FMTO=T,
|          //          IMSID=,SWAP=,DBRC=,IRLM=,IRLMNM=,
|          //          BKO=N,IQB=,SSM=,APARM=,
|          //          RGN=4M,
|          //          SOUT=A,LOGT=2400,SYS2=,
|          //          LOCKMAX=,GSGNAME=,TMINAME=,
|          //          RRS=N,IMSPLEX=
|          //G        EXEC PGM=DFSRR00,REGION=&RGN,
|          //          PARM=(DLI,&MBR,&PSB,&BUF,
|          //          &SPIE&TEST&EXCPVR&RST,&PRLD,
|          //          &SRCH,&CKPTID,&MON,&LOGA,&FMTO,
|          //          &IMSID,&SWAP,&DBRC,&IRLM,&IRLMNM,
|          //          &BKO,&IOB,&SSM,'&APARM',
|          //          &LOCKMAX,&GSGNAME,&TMINAME,
|          //          &RRS,&IMSPLEX)
|          //STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
|          //          DD DSN=IMS.&SYS2.PGMLIB,DISP=SHR
|          //DFSRESLB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
|          //IMS      DD DSN=IMS.&SYS2.PSBLIB,DISP=SHR
|          //          DD DSN=IMS.&SYS2.DBDLIB,DISP=SHR
|          //PROCLIB DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
|          //IEFRDER DD DSN=IMSLOG,DISP=(,KEEP),VOL=(,,99),
|          //          UNIT=(&LOGT,,DEFER),
|          //          DCB=(RECFM=VB,BLKSIZE=4096,
|          //          LRECL=4092,BUFNO=2)
|          //IEFRDER2 DD DSN=IMSLOG2,DISP=(,KEEP),VOL=(,,99),
|          //          UNIT=(&LOGT,,DEFER,SEP=IEFRDER),
|          //          DCB=(RECFM=VB,BLKSIZE=4096,
|          //          LRECL=4092,BUFNO=2)
|          //SYSUDUMP DD SYSOUT=&SOUT,
|          //          DCB=(RECFM=FBA,LRECL=121,BLKSIZE=605),
|          //          SPACE=(605,(500,500),RLSE,,ROUND)
|          //IMSMON  DD DUMMY

```

Figure 20. Procedure for an Offline DL/I Batch Processing Program. This procedure is for DL/I batch jobs using PSB and DBD libraries.

#### Notes:

- Parameters in parentheses are positional.
- The IEFRDER statement is not required if the job does not declare database-update intent.

For a job step declaring database-update intent, DD DUMMY can be specified if the job step is not using DBRC. This is valid where an image copy of the database is taken before the update job step.

Log initialization calculates the smallest value necessary for logical record length. If the JCL logical record length value is larger than the calculated value, the JCL value is used; otherwise, log initialization uses the calculated value for logical record length and adds 4 for the block size.

If multiple volumes are required for the system log, a volume count value should be specified in the VOL parameter of the DD statement.

If the IBM 3480 tape drive is used for the IMS log data set, IMS forces tape write mode (DCB=OPCD=W). The default on the 3480 is to serve as a buffer for the write so that IMS cannot detect when the write is actually performed. If a power failure occurs after a log record is written to the 3480, and the database is

updated but the log record is not yet written to tape, database integrity is lost. Tape write mode is forced for the log in batch and for GSAM data sets.

## DLIBATCH Parameters

Table 27 lists the parameters that are valid for the DLIBATCH procedure. See “Parameter Descriptions for IMS Supplied Procedures” on page 255.

You can specify these parameters in the PARM1= and PARM2= parameters.

*Table 27. Valid parameters for the DLIBATCH procedure*

APARM=	MBR= <sup>1</sup>
BKO=	MON=
BUF=	PRLD=
CKPTID=	PSB=
DBRC=	RGN=
EXCPVR= <sup>1</sup>	RGSUF= <sup>2</sup>
FMTO=	RST= <sup>1</sup>
GSGNAME=	SOUT=
IMSID=	SPIE= <sup>1</sup>
IMSPLEX=	SRCH=
OPB=	SSM=
IRLM=	SWAP=
IRLMNM=	SYS2=
LOCKMAX=	TEST= <sup>1</sup>
LOGA=	TMINAME=
LOGT=	

### Notes:

1. This parameter cannot be specified in the PARM1= and PARM2= parameters.
2. There is no default when specified in the batch procedure.

## DLIBATCH DD Statements

The following DD statements are valid for the DLIBATCH procedure. See “DD Statement Descriptions” on page 303.

DFSHALDB DD <sup>1</sup>	IMSLOGR DD
DFSRESLB DD	IMSMON DD
DFSSTAT DD <sup>1</sup>	PROCLIB DD
DFSVSAMP DD	STEPLIB DD
IEFRDER DD	SYSABEND DD
IEFRDER2 DD <sup>1</sup>	SYSHALDB DD <sup>1</sup>
IMS DD	SYSUDUMP DD

**Note:**<sup>1</sup> This statement is optional.

---

## DLISAS Procedure

The DLISAS procedure initializes a DL/I separate address space (DLISAS).

**Related Reading:** For information on LSO=S see “DL/I Address Space” in the *IMS Version 9: Administration Guide: System*.

The following parameters in the IMS procedure are applicable with the LSO=S option: DLINM, CSAPSB, and DLIPSB.



DLINM specifies the partitioned data set (PDS) member name that IMS uses for the DLISAS procedure. The following list shows how the member name is assigned and how the name can be overridden.

1. DLISAS is the default name
2. The value specified on the DLINM= keyword in the IMSCTRL macro overrides the default name
3. The value specified on the DLINM= keyword in the DFSPBxxx PROCLIB member overrides the value specified in the IMSCTRL macro
4. The value specified on the DLINM= keyword in JCL EXEC parameter overrides the value specified in the DFSPBxxx PROCLIB member

As specified by the DLINM parameter on the IMSCTRL macro, the cataloged procedure (for the DLISAS) is stored in the IMS.PROCLIB member during system definition. This member must be copied to SYS1.PROCLIB.

CSAPSB and DLIPSB specify the PSB pool sizes and override the values specified with the SASPSB parameter on the BUFPOOLS system definition macro.

**Automatic initiation:** An internally defined z/OS START command is issued for the DLINM value and is coded to override the PARM=(DLS,&IMSID) procedure statement. The override defines the IMSID of the subsystem issuing the START, allowing a generic procedure to be used by multiple IMS control regions.

IMS does not issue the IMS READY message until the DLISAS successfully initiates. If the started DLISAS procedure is in error, you should correct and restart it.

When the DL/I address space connects to the IMS control region, the DLISAS procedure name is compared with the DLINM value. If they differ, the DLISAS abends.

**Manual initiation:** You can supply a START command (in member=DLINM of a PDS defined in the control region JCL //PROCLIB) instead of using the internally defined z/OS START command. Columns 1-5 must be "START", or message DFS1930I is issued and the default process is performed.

Here are two types of START commands that you can specify:

- START IMSRDR,MBR=DLISAS: the specified PDS member is read in as a job (rather than as a started task) through the z/OS internal reader.
- START DLIAA,PARM=(DLS,IMSA): DL/I executes as a started task and DLIAA connects to IMSA.

For active and inactive ACBLIBs, the data set names and their concatenation order must be identical (same DSN and VOLSER) in the DLISAS procedure and in the IMS procedure.

IMS.PROCLIB must be defined in the DLISAS procedure and in the IMS procedure. Sometimes the same DLINM member is read from IMS.PROCLIB by both the control region and DLISAS.

## RACF and Resource Protection

If the resources (DL/I databases, for example) are RACF-protected, the user ID associated with DLISAS procedure must be authorized to access them.

**Related Reading:** For more information on how to authorize the user ID, see the *IMS Version 9: Administration Guide: System*.

## DLISAS Procedure Description

The DLISAS procedure is shown in Figure 21.

```

|          //          PROC RGN=64M,DPTY='(14,15)',SOUT=A,
|          //          IMSID=SYS3,SYS2=
|          //IEFPROC EXEC PGM=DFSMVRC0,REGION=&RGN,
|          //          DPRTY=&DPTY,PARM=(DLS,&IMSID)
|          //*****
|          //*
|          //STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
|          //PROCLIB DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
|          //***** ACBLIB *****
|          //*
|          //* THE SPECIFICATION OF THE ACBLIB DATASETS
|          //* IN THE DLI/SAS REGION PROCEDURE MUST
|          //* CORRESPOND EXACTLY WITH THE SPECIFICATION
|          //* IN THE CONTROL REGION JCL
|          //*
|          //IMSACBA DD DSN=IMS.&SYS2.ACBLIBA,DISP=SHR
|          //IMSACBB DD DSN=IMS.&SYS2.ACBLIBB,DISP=SHR
|          //SYSUDUMP DD SYSOUT=&SOUT
|          //SYSABEND DD SYSOUT=&SOUT
|          //***** DATA BASE DD CARDS *****
|          //*
|          //* USER MAY OPTIONALLY SUPPLY THE DD STATEMENTS
|          //* FOR THE ON-LINE DATA BASES TO BE
|          //* INSERTED HERE PRIOR TO ATTEMPTING
|          //* AN ON-LINE SYSTEM EXECUTION USING
|          //* THIS PROCEDURE.
|          //* IF NO DD STATEMENTS ARE SUPPLIED FOR
|          //* A DATA BASE, IMS ASSUMES THAT THIS
|          //* DATA BASE HAS BEEN DESCRIBED THROUGH
|          //* THE DFSMDA MACRO.

```

Figure 21. Procedure to Initialize a DL/I Separate Address Space

## DLISAS Parameters

The following parameters are valid for the DLISAS procedure. See “Parameter Descriptions for IMS Supplied Procedures” on page 255.

DPRTY=	SOUT=
IMSID=	SYS2=
RGN=	

## DLISAS DD Statements

In addition to the following DD statements, statements for DL/I databases must be in this procedure, and not in the IMS procedure; add statements for data sets representing databases that are not to be dynamically allocated. DD Statements for Fast Path databases remain in the IMS procedure. See “DD Statement Descriptions” on page 303.

IMSACBA DD  
IMSACBB DD  
PROCLIB DD

STEPLIB DD  
SYSUDUMP DD

---

## DXRJPROC Procedure

The DXRJPROC procedure starts the Internal Resource Lock Manager (IRLM). This procedure is supplied on the IRLM distribution tape.

**Related Reading:** For information on obtaining the IRLM from the distribution tape and preparing it for use, refer to *IMS Version 9: Installation Volume 1: Installation Verification*.

When installing the IRLM, a unique copy of DXRJPROC should be created for each IRLM that might run concurrently. The procedure name and parameters should be changed, as required, for each IRLM. Because no default value is provided for the IRLMID parameter, the DXRJPROC procedure must at least be altered to assign a value for IRLMID.

In a configuration in which two systems are sharing data, the two IRLMs must execute concurrently. A typical way to define this configuration is:

- Procedure name IRLM1 with IRLMID=1
- Procedure name IRLM2 with IRLMID=2

For testing purposes, two IRLMs can communicate with one another using XCF while executing concurrently on a single system. A typical way to define this environment is:

- Procedure name JRLM1 with IRLMID=1 and IRLMNM=JRLM
- Procedure name KRLM8 with IRLMID=8 and IRLMNM=KRLM

IRLM has a global deadlock manager. This is designated as the IRLM with the lowest IRLMID. IRLMs in a parallel Sysplex group dynamically readjust the global manager identity as members join and leave the group. If you are concerned about the placement of the global deadlock manager, you can put the lowest IRLMID on a specific processor.

## DXRJPROC Procedure Description

The procedure shown in Figure 22 on page 226 executes IRLM 2.1.

```

//DXRJPROC PROC RGN=3072K,
//          IRLMNM=IRLM,
//          IRLMID=,
//          SCOPE=LOCAL,
//          DEADLOK='5,1',
//          MAXCSA=8,
//          PC=NO,
//          MAXUSRS=,
//          IRLMGRP=
//          LOCKTAB=IRLMT1,
//          TRACE=YES,
//          HASH=,
//          LTE=
//          EXEC PGM=DXRRLM00,DPRTY=(15,15),
//          PARM=(&IRLMNM,&IRLMID,&SCOPE,&DEADLOK,&MAXCSA;
//          &PC,&MAXUSRS,&IRLMGRP,&LOCKTAB,&TRACE,&PGPROT,&LTE),
//          REGION=&RGN.
//STEPLIB DD DSN=SDXRRLM21.SDXRRESL,DISP=SHR
//*
//* The following DUMP dd statement should not be specified unless
//* you are having IRLM STARTUP problems and are not getting the
//* dump needed to diagnosis the problem.
//*
//SYSABEND DD SYSOUT=A

```

Figure 22. Procedure to Start IRLM 2.1

## DXRJPROC Parameter Descriptions for IRLM 2.1

The following parameters are valid for the DXRJPROC when you use IRLM 2.1. See "Parameter Descriptions for IMS Supplied Procedures" on page 255.

DEADLOK=	MAXCSA=
HASH=	MAXUSRS=
IRLMGRP=	PC=
IRLMID=	PGPROT=
IRLMNM=	SCOPE=
LOCKTAB=	TRACE=
LTE=	

## DXRJPROC DD Statements for IRLM 2.1

The following DD statements are valid for the DXRJPROC when you are using IRLM 2.1. See "DD Statement Descriptions" on page 303.

STEPLIB DD

SYSABEND DD

---

## FDR Procedure

The FDR procedure executes a Fast Database Recovery (FDBR) address space.

## FDR Procedure Description

The FDR procedure shown in the example executes a Fast Database Recovery (FDBR) address space:

```

//      PROC RGN=64M,SOUT=A,DPTY='(14,15)',
//      SYS=,SYS2=,
//      RGSUF=IMS,PARM1=,PARM2=
//IEFPROC EXEC PGM=DFSMVRC0,DPRTY=&DPTY,
//      REGION=&RGN,
//      PARM='FDR,&RGSUF,&PARM1,&PARM2'
//*
//*
//* THE MEANING AND MAXIMUM SIZE OF EACH PARAMETER
//* IS AS FOLLOWS:
//*
//***** CONTROL REGION SPECIFICATIONS *****
//*****
//*   RGSUF XXX EXEC PARM DEFAULT BLOCK SUFFIX FOR
//*   MEMBER DFSPBXXX.
//*****
//*
//*   PARM1 , PARM2 PARAMETERS BOTH ARE USED TO SPECIFY
//*   CHARACTER STRINGS THAT CONTAIN IMS KEYWORD
//*   PARAMETERS. I.E. PARM1='AUTO=Y,PST=222,RES=Y'
//*
//* ALL OF THE VALID IMS KEYWORD PARAMETERS
//* ARE DESCRIBED BELOW
//*****
//*   FMTO          T = ONLINE FORMATTED DUMP WITH
//*                 STORAGE IMAGE DELETIONS.
//*                 OFFLINE SDUMPS PERMITTED FOR
//*                 NON-IMS TERMINATING ERRORS.
//*   P = FULL ONLINE FORMATTED DUMP.
//*                 OFFLINE SDUMPS PERMITTED FOR
//*                 NON-IMS TERMINATING ERRORS.
//*   F = FULL ONLINE FORMATTED DUMP.
//*                 OFFLINE SDUMPS SUPPRESSED FOR
//*                 NON-IMS TERMINATING ERRORS.
//*   N = NO FORMATTED DUMP, NO OFFLINE
//*                 DUMP. OFFLINE SDUMPS PERMITTED
//*                 FOR NON-IMS TERMINATING ERRORS
//*   Z = NO FORMATTED DUMP, NO OFFLINE
//*                 DUMP. OFFLINE SDUMPS
//*                 SUPPRESSED FOR NON-IMS
//*                 TERMINATING ERRORS.
//*   (DEFAULT)   D = OFFLINE DUMP, OR ONLINE FORMAT-
//*                 TED DUMP WITH STORAGE IMAGE
//*                 DELETIONS IF OFFLINE DUMPING
//*                 FAILS. OFFLINE SDUMPS
//*                 PERMITTED FOR NON-IMS
//*                 TERMINATING ERRORS.
//*   X = OFFLINE DUMP, OR ONLINE FORMAT-
//*                 TED DUMP WITH STORAGE IMAGE
//*                 DELETIONS IF OFFLINE DUMPING
//*                 FAILS. OFFLINE SDUMPS
//*                 SUPPRESSED FOR NON-IMS
//*                 TERMINATING ERRORS.
//*   M = OFFLINE DUMP, ONLINE IMS DUMP
//*                 FORMATTING NOT PERMITTED.
//*                 OFFLINE SDUMPS PERMITTED FOR
//*                 NON-IMS TERMINATING ERRORS.
//*   R = OFFLINE DUMP, ONLINE IMS DUMP
//*                 FORMATTING NOT PERMITTED.
//*                 OFFLINE SDUMPS SUPPRESSED FOR
//*                 NON-IMS TERMINATING ERRORS.

```

```

|      /** IMSID XXXX IMS SUBSYSTEM IDENTIFIER
|      /** ARMST X    Y = ALLOW MVS ARM TO RESTART
|      /**           N = ARM NOT RESTART IMS
|      /** IRLMNM XXXX IRLM SUBSYSTEM NAME
|      /** WADS X    SINGLE OR DUAL WADS,S=SINGLE,D=DUAL
|      /** ARC  XX   AUTOMATIC ARCHIVE.
|      /**           0 = NOT AUTOMATIC
|      /**           1-99 = AUTOMATIC
|      /** UHASH XXXXXXXX USER HASH MODULE NAME
|      /** DESC  XX   MSG DESC CODE
|      /** MCS   (XX,XX) MSG ROUTE CODES
|      /** SVC2  XXX  TYPE 2 SVC NUMBER
|      /******* FAST PATH PARAMETERS *****
|      /**
|      /** BSIZ  XXXXX DATA BASE BUFFER SIZE
|      /** DBBF  XXXXX NUMBER OF DATABASE BUFFERS
|      /** LGNR  XX   NUMBER OF LOG ENTRIES IN DEDB BUFFERHEADER
|      /******* STORAGE POOL VALUES IN K, M OR G *****
|      /**
|      /** PSB   XXXXXX PSB POOL SIZE - NON DLISAS
|      /** DMB   XXXXXX DMB POOL SIZE
|      /** WKAP  XXXXXX WORKING STORAGE BUFFER POOL SIZE
|      /** DBWP  XXXXXX DATABASE WORK POOL SIZE
|      /** CSAPSB XXXXXX DLISAS: CSA PSB POOL SIZE
|      /** DLIPSB XXXXXX DLISAS: DLI PSB POOL SIZE
|      /**
|      /******* MEMBER SUFFIXES *****
|      /**
|      /** SUF   X    LAST CHARACTER OF CTL PROGRAM LOAD
|      /**           MODULE MEMBER NAME
|      /** VSPEC XX   2 CHARACTER BUFFER POOL SPEC MODULE SUFFIX
|      /** SPM   XX   STG POOL MGR PROCLIB MEMBER SUFFIX
|      /**
|      /*******
|      /**
|      /**STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
|      /**PROCLIB DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
|      /**JCLOUT DD SYSOUT=(A,INTRDR)
|      /**JCLPDS DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
|      /**
|      /******* DASD LOGGING DD CARDS *****
|      /** THE FOLLOWING DD CARDS DESCRIBE THE DASD LOGGING
|      /** OLDS AND WADS. THESE CARDS ARE FOR EXAMPLE ONLY.
|      /** ALL OLDS AND WADS DATA SETS MAY BE DYNAMICALLY
|      /** ALLOCATED. DD CARDS ARE NOT REQUIRED.
|      /** THE OLDS AND WADS TO BE USED DURING STARTUP MUST
|      /** BE SPECIFIED VIA OLDSDEF AND WADSDEF CONTROL
|      /** STATEMENTS IN THE DFSVSMXX MEMBER OF IMS PROCLIB.
|      /** THE ACTUAL SELECTION OF OLDS AND WADS MUST BE
|      /** TAILORED TO INSTALLATION REQUIREMENTS. THE OLDS
|      /** AND WADS MUST BE PREDEFINED BY A SET UP JOB.
|      /** THE BLOCK SIZE OF ALL OLDS MUST BE THE SAME.
|      /** THE BLOCK SIZE AND DEVICE TYPE OF ALL WADS MUST
|      /** BE THE SAME. AT LEAST 3 PRIMARY OLDS AND 1 WADS
|      /** MUST BE AVAILABLE FOR STARTUP. THE BLOCK SIZE
|      /** SHOULD NOT BE SPECIFIED IN THIS JCL. THE LOGGER
|      /** WILL GET THE BLOCK SIZE FROM THE VTOC.
|      /**
|      /**DFSOLP00 DD DSN=IMS.&SYS.OLP00,DISP=SHR
|      /**DFSOLP01 DD DSN=IMS.&SYS.OLP01,DISP=SHR
|      /**DFSOLP02 DD DSN=IMS.&SYS.OLP02,DISP=SHR

```

```

| //DFSOLP03 DD DSN=IMS.&SYS.OLP03,DISP=SHR
| //DFSOLP04 DD DSN=IMS.&SYS.OLP04,DISP=SHR
| //DFSOLP05 DD DSN=IMS.&SYS.OLP05,DISP=SHR
| /*
| //DFSOLS00 DD DSN=IMS.&SYS.OLS00,DISP=SHR
| //DFSOLS01 DD DSN=IMS.&SYS.OLS01,DISP=SHR
| //DFSOLS02 DD DSN=IMS.&SYS.OLS02,DISP=SHR
| //DFSOLS03 DD DSN=IMS.&SYS.OLS03,DISP=SHR
| //DFSOLS04 DD DSN=IMS.&SYS.OLS04,DISP=SHR
| //DFSOLS05 DD DSN=IMS.&SYS.OLS05,DISP=SHR
| /*
| //DFSWADS0 DD DSN=IMS.&SYS.WADS0,DISP=SHR
| //DFSWADS1 DD DSN=IMS.&SYS.WADS1,DISP=SHR
| /*
| //IMSACBA DD DSN=IMS.&SYS2.ACBLIBA,DISP=SHR
| //IMSACBB DD DSN=IMS.&SYS2.ACBLIBB,DISP=SHR
| //MODBLKSA DD DSN=IMS.&SYS2.MODBLKSA,DISP=SHR
| //MODBLKSB DD DSN=IMS.&SYS2.MODBLKSB,DISP=SHR
| //MODSTAT DD DSN=IMS.&SYS.MODSTAT,DISP=SHR
| //***** SYSTEM STATEMENTS *****
| /*
| //SYSUDUMP DD SYSOUT=&SOUT,
| //          DCB=(LRECL=125,RECFM=FBA,BLKSIZE=3129),
| //          SPACE=(6050,300,,ROUND)
| //IMSRDS DD DSN=IMS.&SYS.RDS,DISP=SHR
| /*
| //***** DATA BASE DD CARDS *****
| /*
| /* USER MAY OPTIONALLY SUPPLY THE DD STATEMENTS
| /* FOR THE ON-LINE DATA BASES TO BE
| /* INSERTED HERE PRIOR TO ATTEMPTING
| /* AN ON-LINE SYSTEM EXECUTION USING
| /* THIS PROCEDURE.
| /* IF NO DD STATEMENTS ARE SUPPLIED FOR
| /* A DATA BASE, IMS ASSUMES THAT THIS
| /* DATA BASE HAS BEEN DESCRIBED THROUGH
| /* THE DFSMDA MACRO.
| /* IF THE USER WILL BE EXECUTING WITH THE DL/I
| /* SAS OPTION, THESE DD STATEMENTS SHOULD BE ADDED
| /* TO THE DLISAS PROCLIB MEMBER OR DESCRIBED
| /* THROUGH THE DFSMDA MACRO.
| /*
| //***** DBRC RECON DD CARDS *****
| /*
| /* USER MAY OPTIONALLY SUPPLY THE DD CARDS
| /* REQUIRED FOR THE DBRC RECON DATA SET.
| /* IF NO DD STATEMENTS ARE SUPPLIED FOR RECON
| /* DATASETS, IMS ASSUMES THAT THE DATASETS
| /* HAVE BEEN DESCRIBED THROUGH THE DFSMDA MACRO.

```

## FDR Procedure Parameters

The following parameters are valid for the FDR procedure. See “Parameter Descriptions for IMS Supplied Procedures” on page 255.

	ARC=	MCS=
	ARMRST=	PSB=
	BSIZ=	RGN=
	CSAPSB=	RGSUF=
	CSLG=	SOUT=
	DBBF=	SPM=
	DBWP=	SUF=
	DESC=	SVC2=
	DLIPSB=	SYS=
	DMB=	SYS2=
	DPRTY=	UHASH=
	FMTO=	VSPEC=
	IMSID=	WADS=
	IRLMNM=	WKAP=
	LGNR=	

For the FDR procedure, the sum of the values for the CSAPSB and DLIPSB defines the PSB pool size. If PSB is also specified, the larger value (PSB or the sum of CSAPSB and DLIPSB) is used.

## FDR Procedure DD Statements

The following DD statements are valid for the FDR procedure. In addition to the following DD statements, add statements for data sets that are not to be dynamically allocated. See “DD Statement Descriptions” on page 303.

	DFSOLPn DD	JCLPDS DD
	DFSOLSn DD	MODBLKSA DD
	DFSWADSn DD	MODBLKSB DD
	IMSACBA DD	MODSTAT DD
	IMSACBB DD	PROCLIB DD
	IMSRDS DD	STEPLIB DD
	JCLOUT DD	SYSUDUMP DD

---

## FPUTIL Procedure

| The FPUTIL Procedure executes the Fast Path utility programs with Data Entry Database (DEDDB) online.

## FPUTIL Procedure Description

The procedure shown in Figure 23 on page 231 executes the Fast Path utility programs with Data Entry Database (DEDDB) online.



```

| //FPUTIL PROC SOUT=A,RGN=4M,SYS2=,
| //          DBD=,REST=00,DIRCA=002,
| //          PRLD=,IMSID=,AGN=,SSM=,ALTID=
| //FPU EXEC PGM=DFSRR00,REGION=&RGN,
| //          PARM=(IFP,&DBD,DBF#FPU0,&REST,00,,1,
| //          &DIRCA,&PRLD,0,,, &IMSID,&AGN,&SSM,,
| //          &ALTID)
| //STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
| //PROCLIB DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
| //SYSPRINT DD SYSOUT=&SOUT
| //SYSUDUMP DD SYSOUT=&SOUT,
| //          DCB=(LRECL=121,RECFM=VBA,BLKSIZE=3129),
| //          SPACE=(125,(2500,100),RLSE,,ROUND)

```

Figure 23. Procedure to Execute Fast Path Utility Programs

**Note:** Parameters in parentheses are positional.

NBA= and OBA= are not valid parameters for this procedure. The following default values are used: NBA=7 and OBA=0.

## FPUTIL Procedure Parameters

The following parameters are valid for the FPUTIL procedure. See “Parameter Descriptions for IMS Supplied Procedures” on page 255.

AGN=	REST=
ALTID=	RGN=
DBD=	SOUT=
DIRCA=	SSM=
IMSID=	SYS2=
PRLD=	

## FPUTIL Procedure DD Statements

The following DD statements are valid for the FPUTIL procedure. See “DD Statement Descriptions” on page 303.

PROCLIB DD	SYSPRINT DD
STEPLIB DD	SYSUDUMP DD

---

## IMS Procedure

The IMS Procedure is an online execution procedure to initialize the DB/DC environment.

**RACF and Resource Protection:** If the resources (for example, DL/I databases) are RACF protected, the user ID associated with the IMS procedure must be authorized to access them.

**Related Reading:** For more information on how to authorize the user ID, see *IMS Version 9: Administration Guide: System*.

**DLISAS Procedure:** If you specify LSO=S in the IMS procedure, thereby automatically invoking the DLISAS procedure, the DD statements for the DL/I databases must be in the DLISAS procedure, and not in the IMS procedure. Dynamic allocation members remain in the STEPLIB library.

**Program Specification Block (PSB) Pools:** If you do not specify LSO=S in the IMS procedure, one PSB pool is created, as specified by the PSB parameter on the BUFPOOLS macro. If you do specify LSO=S, two PSB pools are created and the PSB parameter is ignored. The pool sizes are specified with the SASPSB parameter on the BUFPOOLS system definition macro, and overridden with the CSAPSB and DLIPSB parameters on the IMS procedure.

**Related Reading:** For an overview of information about the DL/I address space options, see *IMS Version 9: Administration Guide: System*.

The control region procedure contains the following parameters:

```
IMSPROC &RGN=,&SOUT=,&DPTY=,&LOGT=,&RGSUF=000,
        &PARM1=,&PARM2=
```

The PARM2 parameter is needed because z/OS:

- Allows up to 100 characters in the EXEC parameter area
- Does not allow symbolic parameters to be continued on next record
- Does not allow more than 66 characters for each keyword option phrase on the START command.

## IMS Procedure Description

The following procedure executes a DB/DC online control program.

```
| //      PROC RGN=64M,SOUT=A,DPTY='(14,15)',
| //      SYS=,SYS1=,SYS2=,
| //      LOGT=2400,
| //      RGSUF=IMS,PARM1=,PARM2=
| //IEFPROC EXEC PGM=DFSMVRC0,DPRTY=&DPTY,
| //      REGION=&RGN,
| //      PARM='CTL,&RGSUF,&PARM1,&PARM2'
| //*
| //*
| //* THE MEANING AND MAXIMUM SIZE OF EACH PARAMETER
| //* IS AS FOLLOWS:
| //*
| //***** CONTROL REGION SPECIFICATIONS *****
| //*****
| //*   RGSUF   XXX   EXEC PARM DEFAULT BLOCK SUFFIX FOR
| //*           MEMBER DFSPBXXX.
| //*****
| //*
| //*   PARM1 , PARM2 PARAMETERS BOTH ARE USED TO SPECIFY
| //*   CHARACTER STRINGS THAT CONTAIN IMS KEYWORD
| //*   PARAMETERS. I.E. PARM1='AUTO=Y,PST=222,RES=Y'
| //*
| //*   ALL OF THE VALID IMS KEYWORD PARAMETERS
| //*   ARE DESCRIBED BELOW
| //*****
| //*   APPLID1 XXXXXXXX VTAM APPLID OF ACTIVE IMS SYSTEM
| //*   APPLID2 XXXXXXXX VTAM APPLID OF XRF ALTERNATE SYSTEM
| //*   APPLID3 XXXXXXXX VTAM APPLID OF RSR TRACKING SYSTEM
| //*   RES      X      BLOCK RESIDENT (N = NO, Y = YES)
| //*   FRE      XXXXX  NUMBER OF FORMAT REQUEST ELEMENTS
| //*   PST      XXX    NUMBER OF PST'S PERMANENTLY ALLOC
| //*   MAXPST   XXX    MAXIMUM NUMBER OF PST'S
| //*   SAV      XXX    NUMBER OF DYNAMIC SAVE AREA SETS
| //*   SRCH    X      MODULE SEARCH INDICATOR FOR DIRECTED LOAD
| //*
| //*           0 = STANDARD SEARCH
```

```

|          /**          1 = SEARCH JPA AND LPA BEFORE PDS
|          /** SOD      X  1 CHARACTER SYSOUT CLASS
|          /** VAUT     X  VTAM AUTH PATH OPTION (1=YES,0=NO)
|          /** FMTO          T = ONLINE FORMATTED DUMP WITH
|          /**          STORAGE IMAGE DELETIONS.
|          /**          OFFLINE SDUMPS PERMITTED FOR
|          /**          NON-IMS TERMINATING ERRORS.
|          /**          P = FULL ONLINE FORMATTED DUMP.
|          /**          OFFLINE SDUMPS PERMITTED FOR
|          /**          NON-IMS TERMINATING ERRORS.
|          /**          F = FULL ONLINE FORMATTED DUMP.
|          /**          OFFLINE SDUMPS SUPPRESSED FOR
|          /**          NON-IMS TERMINATING ERRORS.
|          /**          N = NO FORMATTED DUMP, NO OFFLINE
|          /**          DUMP. OFFLINE SDUMPS PERMITTED
|          /**          FOR NON-IMS TERMINATING ERRORS
|          /**          Z = NO FORMATTED DUMP, NO OFFLINE
|          /**          DUMP. OFFLINE SDUMPS
|          /**          SUPPRESSED FOR NON-IMS
|          /**          TERMINATING ERRORS.
|          /** (DEFAULT) D = OFFLINE DUMP, OR ONLINE FOR-
|          /**          MATTED DUMP WITH STORAGE IMAGE
|          /**          DELETIONS IF OFFLINE DUMPING
|          /**          FAILS. OFFLINE SDUMPS
|          /**          PERMITTED FOR NON-IMS
|          /**          TERMINATING ERRORS.
|          /**          X = OFFLINE DUMP, OR ONLINE FOR-
|          /**          MATTED DUMP WITH STORAGE IMAGE
|          /**          DELETIONS IF OFFLINE DUMPING
|          /**          FAILS. OFFLINE SDUMPS
|          /**          SUPPRESSED FOR NON-IMS
|          /**          TERMINATING ERRORS.
|          /**          M = OFFLINE DUMP, ONLINE IMS DUMP
|          /**          FORMATTING NOT PERMITTED.
|          /**          OFFLINE SDUMPS PERMITTED FOR
|          /**          NON-IMS TERMINATING ERRORS.
|          /**          R = OFFLINE DUMP, ONLINE IMS DUMP
|          /**          FORMATTING NOT PERMITTED.
|          /**          OFFLINE SDUMPS SUPPRESSED FOR
|          /**          NON-IMS TERMINATING ERRORS.
|          /** AUTO      X  Y = AUTOMATIC RESTART DESIRED
|          /**          N = NO AUTOMATIC RESTART
|          /** IMSID     XXXX IMS SUBSYSTEM IDENTIFIER
|          /** NLXB      XXX  # ADD'L LXBS FOR MSC VTAM
|          /** LSO       X  Y = DL/I LOCAL STORAGE OPTION ON
|          /**          S = DLI/SAS OPTION
|          /**
|          /** APPC      X  Y = ACTIVATE APPC/IMS
|          /**          N = DO NOT ACTIVATE APPC/IMS
|          /** LTERM     X  Y = LTERM USED IN DFSAPPC PROCESS
|          /**          N = LTERM NOT USED IN DFSAPPC
|          /**          PROCESS
|          /** ARMRST   X  Y = ALLOW MVS ARM TO RESTART
|          /**          N = ARM NOT RESTART IMS
|          /** RRS       X  Y = ENABLE PROT CONV SUPPORT
|          /**          N = DISABLE PROT CONV SUPPORT
|          /** IRLM      X  Y = YES, N = NO
|          /** IRLMNM   XXXX IRLM SUBSYSTEM NAME
|          /** SSM       XXXX EXT SUBSYSTEM PROCLIB MEMBER ID
|          /** WADS      X  SINGLE/DUAL WADS,S=SINGLE,D=DUAL

```

```

|          /** ARC      XX    AUTOMATIC ARCHIVE.
|          /**                0 = NOT AUTOMATIC
|          /**                1-99 = AUTOMATIC
|          /** UHASH    XXXXXXXX USER HASH MODULE NAME
|          /** DBRCNM   XXXXXXXX DBRC PROCLIB MEMBER NAME
|          /** DLINM    XXXXXXXX DL/I PROCLIB MEMBER NAME
|          /** PRDR     XXXXXXXX IMSRDR PROCLIB MEMBER NAME
|          /** FESTIM   XXXX    FRONTENDSWITCH TIMEOUT (SECONDS)
|          /** RECASZ   XXXXX   RECEIVE ANY BUFFER SIZE
|          /** PIMAX    XXXXXX  ENQ/DEQ POOL MAXIMUM BYTES
|          /** PIINCR   XXXXXX  ENQ/DEQ POOL INCREMENT
|          /** RECA     XXX     NUMBER OF RECEIVE ANY BUFFERS
|          /** CRC      X      COMMAND RECOGNITION CHARACTER
|          /** TSR      X      U = UTC TIME
|          /**                L = LOCAL TIME (DEFAULT)
|          /** YEAR4    X      N = 2-DIGIT DATE (DEFAULT)
|          /**                Y = 4-DIGIT DATE
|          /** DC       XXX     DC PROC MEMBER SUFFIX IN
|          /**                IMS.PROCLIB
|          /**                DEFAULT VALUE IS 000
|          /** CPLOG    XXXXXX  CHECKPOINT LOG INTERVAL
|          /**                OR
|          /** CPLOG    XXM     CHECKPOINT LOG INTERVAL
|          /** PASSWD1  XXXXXXXX VTAM ACB PASSWORD
|          /** ORSMBR   XX      SUFFIX FOR ORS MEMBER
|          /** IMSGROUP XXXX   4 CHAR USER SPEC NAME
|          /** DESC     XX      MSG DESC CODE
|          /** MCS      (XX,XX) MSG ROUTE CODES
|          /** SVC2     XXX     TYPE 2 SVC NUMBER
|          /******* OTMA PARAMETERS *****
|          /**
|          /** OTMA     X      Y = OTMA ENABLED
|          /**                N = OTMA NOT ENABLED
|          /**                DEFAULT VALUE IS N
|          /** OTMANM   XXXXXXXX IMS OTMA XCF MEMBER NAME
|          /** GRNAME    XXXXXXXX OTMA XCF GROUP NAME
|          /**                NO DEFAULT VALUE
|          /** GRSNAME  XXXXXXXX GENERIC RESOURCE GROUP
|          /**                NAME
|          /**                NO DEFAULT VALUE
|          /**
|          /******* SECURITY PARAMETERS *****
|          /**
|          /** AOIS     X      ICMD SECURITY OPTION
|          /** AOI1     X      CMD SECURITY OPTION
|          /**                A = ALL
|          /**                N = NONE
|          /**                C = DFSCCMD0 EXIT
|          /**                R = RACF
|          /**                S = SMU
|          /** TCORACF  X      TCO RACF SECURITY OPTION
|          /**                Y = YES
|          /**                N = NO
|          /** APPCSE   X      C = APPC RACF SECURITY IS CHECK
|          /**                F = APPC RACF SECURITY IS FULL
|          /**                N = APPC RACF SECURITY IS NONE
|          /**                P = APPC RACF SECURITY IS PROFILE
|          /** CMDMCS   X      MCS/EMCS COMMAND OPTION
|          /**                N=COMMANDS NOT ALLOWED WITH CRC
|          /**                Y=ALL COMMANDS ALLOWED WITH CRC
|          /**                R=RACF COMMAND SECURITY

```

```

|          /**          C=DFSCCMD0 COMMAND SECURITY
|          /**          B=RACF AND DFSCCMD0 CMD SEC
|          /**  ISIS    X    N = NO RESOURCE ACCESS SECURITY
|          /**          R = RACF RESOURCE ACCESS SECURITY
|          /**          C = RACF RESOURCE ACCESS SECURITY
|          /**          A = RACF RESOURCE ACCESS SECURITY
|          /**          0 = NO AGN SECURITY
|          /**          1= APPLICATION GROUP NAME SECURITY
|          /**          2= APPLICATION GROUP NAME SECURITY
|          /**  RCF     X    RACF USED FOR TRANS. AND SIGNON
|          /**          A = Y + S, Y = T + C, S = S + C.
|          /**          B = 'A' + SVST TABLE NOT LOADED
|          /**          R = 'S' + SVST TABLE NOT LOADED
|          /**  RVFY    X    RACF REVERIFY OPTION
|          /**          Y = YES, N = NO
|          /**  SGN     X    SIGNON AUTHORIZATION CHECKING
|          /**          F = MTO CANNOT NEGATE LOADING
|          /**          SIGNON VERIFICATION SECURITY
|          /**          TABLE(SVST) FROM IMS.MATRIX
|          /**          Y = SVST LOADED UNLESS OVERRIDEN
|          /**          BY MTO
|          /**          N = SVST NOT LOADED UNLESS
|          /**          OVERRIDEN BY MTO
|          /**          M = SINGLE USERID CAN SIGNON
|          /**          TO MULTIPLE STATIC TERMINALS
|          /**          G = 'F' + 'M'
|          /**          Z = 'Y' + 'M'
|          /**          D = 'F' + SVST TABLE NOT LOADED
|          /**          E = 'G' + SVST TABLE NOT LOADED
|          /**          W = 'Y' + SVST TABLE NOT LOADED
|          /**          X = 'Z' + SVST TABLE NOT LOADED
|          /**  TRN     X    TRANSACTION AUTHORIZATION CHECKING
|          /**          F = FORCED, Y = YES, N = NO
|          /**          E = 'F' + SVST TABLE NOT LOADED
|          /**          X = 'Y' + SVST TABLE NOT LOADED
|          /**  RCFTCB  XX   NUMBER OF RCF TCB'S
|          /**
|          /******* MESSAGE QUEUE PARAMETERS *****
|          /**
|          /**  EXVR    X    PAGEFIX QMGR BUFFER POOLS
|          /**          (1=YES, 0=NO)
|          /**  QBUF    XXXX  NUMBER OF MESSAGE QUEUE BUFFERS
|          /**  QTL     XXX   QUEUE LOWER THRESHOLD (%)
|          /**  QTU     XXX   QUEUE UPPER THRESHOLD (%)
|          /**
|          /******* SHARED QUEUES PARAMETERS *****
|          /**
|          /**  LGMSGSZ XXXXX LONG MESSAGE SIZE
|          /**  QBUFHITH XXX  MSG QBUF HIGH THESHOLD %
|          /**  QBUFLWTH XXX  MSG QBUF LOW THESHOLD %
|          /**  QBUFMAX  XXXX  MAX NUMBER OF MSG QUEUE BUFFERS
|          /**  QBUFPCTX XXX  % MSG QBUF DYNAMIC EXPAND
|          /**          WHEN QBUFHITH EXCEEDED
|          /**          DEFAULT IS 20%
|          /**  QBUFSZ  XXXXX  SIZE OF MESSAGE QUEUE BUFFERS
|          /**  SHMSGSZ XXXXX  SHORT MESSAGE SIZE
|          /**  SHAREDQ XXX  SQ PROC MEMBER SUFFIX IN
|          /**          IMS.PROCLIB
|          /**          NO DEFAULT VALUE
|          /**
|          /******* XRF PARAMETERS *****

```

```

|      /**
|      /** HSBID  X    XRF SYSTEM ID
|      /**                1 FOR FIRST SYSTEM
|      /**                2 FOR SECOND SYSTEM
|      /**
|      /** HSBMBR XX  SUFFIX FOR XRF MEMBER IN
|      /**                IMS.PROCLIB
|      /**                00 IS DEFAULT
|      /**
|      /** MNPS   XXXXXXXX NAME OF MNPS ACB
|      /**                USERVAR WILL BE IGNORED
|      /** MNPSPW XXXXXXXX MNPS ACB PASSWORD
|      /**
|      /******* FDR PARAMETER *****
|      /**
|      /** FDRMBR XX  SUFFIX FOR FDR MEMBER IN
|      /**                IMS.PROCLIB
|      /**
|      /******* FAST PATH PARAMETERS *****
|      /**
|      /** BSIZ   XXXXX DATA BASE BUFFER SIZE
|      /** OTHR   XXX  NUMBER OF OUTPUT THREADS
|      /** DBFX   XXXXX SYSTEM ALLOCATION OF DATA BASE BUFFERS TO BE
|      /**                FIXED AT START OF 1ST FAST PATH DEP REGION
|      /** DBBF   XXXXX NUMBER OF DATABASE BUFFERS
|      /** DBFP   XXXX  PAGE FIX/FREE ADJUST TIMER
|      /**                0: FIX/FREE AT SCHED/TERM
|      /**                1: ALLOW PAGEFIX ONLY
|      /**                2-3600: SEC PAGEFREE FREQ
|      /** MSDB   X    SUFFIX FOR MSDB MEMBER ON
|      /**                IMS.PROCLIB
|      /** LGNR   XX   NUMBER OF LOG ENTRIES IN DEDB BUFFERHEADER
|      /** EMHL   XXXXX SIZE OF EMH BUFFER IN BYTES
|      /** SVSODR XXXX  SVSO DISASTER RECOVERY OPTIONS
|      /**                NONE: DEFAULT. NO CHANGE TO ERE.
|      /**                AUTO: AREA MARKED RECOV NEEDED IF
|      /**
|      /**                DRRS: AREA MARKED RECOV NEEDED AT
|      /**
|      /**                WTOR: USER OPTION TO MARK AREAS
|      /**
|      /**
|      /** DMHVF  XX   MEGS TO FIX PAGEFIX FOR VSO ERE      DATASPACE
|      /** FPOP  X    DISABLE CTL REG PREOPEN FOR DEDBS
|      /**                BLANK. DEFAULT. PREOPEN OF
|      /**                DEDB AREAS DONE AT CTL REGION
|      /**                INITIALIZATION
|      /**                D: DEDB PREOPEN/PRELOAD IS DIS-
|      /**                ABLED AT CTL REGION INIT
|      /**
|      /******* ETO PARAMETERS *****
|      /**
|      /** ETO    X    Y = EXTENDED TERMINAL OPTION
|      /**                N = NO EXTENDED TERMINAL OPTION
|      /**                M = NO EXTENDED TERMINAL OPTION
|      /**                BUT LOGON USERDATA SUPPORTED
|      /**                FOR STATIC TERMINALS
|      /** ASOT   XXXX  ETO AUTO SIGNOFF TIME
|      /** ALOT   XXXX  ETO AUTO LOGNOFF TIME
|      /** DLQT   XXX   ETO DEAD LETTER QUEUE SIZE
|      /**

```

```

|          //***** RSR PARAMETERS *****
|          //*
|          //*   RSRMBR  XX   SUFFIX FOR RSR MEMBER
|          //*   TRACK   XXX  NO = NO RECOVERY TRACKING DONE
|          //*                               RLT = RECOVERY TRACKING DONE
|          //*                               DLT = DATABASE TRACKING DONE
|          //*   USERVAR  XXXXXXXX USER NAME OF ACTIVE IMS SYSTEM FOR RSR
|          //*
|          //***** HASH TABLE PARAMETERS *****
|          //*
|          //*   LHTS   XXXXX # OF CNT HASH TABLE SLOTS
|          //*   NHTS   XXXXX # OF VTCB HASH TABLE SLOTS
|          //*   UHTS   XXXXX # OF SPQB HASH TABLE SLOTS
|          //*
|          //***** STORAGE POOL VALUES IN K, M OR G *****
|          //*
|          //*   FBP     XXXXXX MESSAGE BUFFER POOL SIZE
|          //*   PSB     XXXXXX PSB POOL SIZE - NON DLISAS
|          //*   DMB     XXXXXX DMB POOL SIZE
|          //*   CIOP    XXXXXX CIOP POOL UPPER LIMIT
|          //*   WKAP    XXXXXX WORKING STORAGE BUFFER POOL SIZE
|          //*   PSBW    XXXXXX PSB WORK POOL SIZE
|          //*   DBWP    XXXXXX DATABASE WORK POOL SIZE
|          //*   CSAPSB  XXXXXX DLISAS: CSA PSB POOL SIZE
|          //*   DLIPSB  XXXXXX DLISAS: DLI PSB POOL SIZE
|          //*   EPCB    XXXXXX EPCB POOL SIZE
|          //*   HIOP    XXXXXX HIOP POOL UPPER LIMIT
|          //*   FPWP    XXXXXX FPWP POOL UPPER LIMIT
|          //*   EMHB    XXXXXX EMHB POOL UPPER LIMIT
|          //*   LUMP    XXXXXX LUMP POOL UPPER LIMIT
|          //*   LUMC    XXXXXX LUMC POOL UPPER LIMIT
|          //*   AOIP    XXXXXX AOI POOL UPPER LIMIT
|          //*
|          //***** MEMBER SUFFIXES *****
|          //*
|          //*   SUF     X    LAST CHARACTER OF CTL PROGRAM LOAD
|          //*                               MODULE MEMBER NAME
|          //*   FIX     XX   2 CHARACTER FIX PROCEDURE MODULE SUFFIX
|          //*   PRLD    XX   2 CHARACTER PROCLIB MEMBER SUFFIX FOR PRELOAD
|          //*   VSPEC   XX   2 CHARACTER BUFFER POOL SPEC MODULE SUFFIX
|          //*   SPM     XX   STG POOL MGR PROCLIB MEMBER SUFFIX
|          //*   CSLG    XXX   CSL GLOBAL MEMBER (DFSCGXXX)
|          //*   DSCT    X    ETO USER DESCRIPTOR TABLE(DFSDSCTX)
|          //*
|          //*****
|          //*
|          //*           DATA DEFINITION STATEMENTS FOLLOW           *
|          //*                                                                 *
|          //*****
|          //*
|          //***** LIBRARY STATEMENTS *****
|          //*
|          //STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
|          //PROCLIB DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
|          //*
|          //***** GENERIC START DEPENDANT REGION *****
|          //*
|          //* IN ORDER TO START A DEPENDENT REGION, MODIFIED
|          //* START-UP JCL IS WRITTEN FROM INTERNAL STORAGE TO
|          //* THE INTERNAL READER.
|          //*

```

```

| //IMSIRD DD SYSOUT=(A,INTRDR)
| //*
| //***** DASD LOGGING STATEMENTS *****
| //*
| //* THE FOLLOWING DD CARDS DESCRIBE THE DASD LOGGING
| //* OLDS AND WADS. THESE CARDS ARE FOR EXAMPLE ONLY.
| //* ALL OLDS AND WADS DATA SETS MAY BE DYNAMICALLY
| //* ALLOCATED. DD CARDS ARE NOT REQUIRED.
| //* THE OLDS AND WADS TO BE USED DURING STARTUP MUST
| //* BE SPECIFIED VIA OLDSDEF AND WADSDEF CONTROL
| //* STATEMENTS IN THE DFSVSMXX MEMBER OF IMS PROCLIB.
| //* THE ACTUAL SELECTION OF OLDS AND WADS MUST BE
| //* TAILORED TO INSTALLATION REQUIREMENTS. THE OLDS
| //* AND WADS MUST BE PREDEFINED BY A SET UP JOB.
| //* THE BLOCK SIZE OF ALL OLDS MUST BE THE SAME.
| //* THE BLOCK SIZE AND DEVICE TYPE OF ALL WADS MUST
| //* BE THE SAME. AT LEAST 3 PRIMARY OLDS AND 1 WADS
| //* MUST BE AVAILABLE FOR STARTUP. THE BLOCK SIZE
| //* SHOULD NOT BE SPECIFIED IN THIS JCL. THE LOGGER
| //* WILL GET THE BLOCK SIZE FROM THE VTOC.
| //*
| //DFSOLP00 DD DSN=IMS.&SYS.OLP00,DISP=SHR
| //DFSOLP01 DD DSN=IMS.&SYS.OLP01,DISP=SHR
| //DFSOLP02 DD DSN=IMS.&SYS.OLP02,DISP=SHR
| //DFSOLP03 DD DSN=IMS.&SYS.OLP03,DISP=SHR
| //DFSOLP04 DD DSN=IMS.&SYS.OLP04,DISP=SHR
| //DFSOLP05 DD DSN=IMS.&SYS.OLP05,DISP=SHR
| //*
| //DFSOLS00 DD DSN=IMS.&SYS.OLS00,DISP=SHR
| //DFSOLS01 DD DSN=IMS.&SYS.OLS01,DISP=SHR
| //DFSOLS02 DD DSN=IMS.&SYS.OLS02,DISP=SHR
| //DFSOLS03 DD DSN=IMS.&SYS.OLS03,DISP=SHR
| //DFSOLS04 DD DSN=IMS.&SYS.OLS04,DISP=SHR
| //DFSOLS05 DD DSN=IMS.&SYS.OLS05,DISP=SHR
| //*
| //DFSWADS0 DD DSN=IMS.&SYS.WADS0,DISP=SHR
| //DFSWADS1 DD DSN=IMS.&SYS.WADS1,DISP=SHR
| //*
| //***** MONITOR STATEMENTS *****
| //*
| //* THE IMSMON DD STATEMENT MUST BE REMOVED IF
| //* THIS DATA SET IS TO BE DYNAMICALLY ALLOCATED.
| //*
| //IMSMON DD DSN=IMS.&SYS1.IMSMON,DISP=(,KEEP),
| // VOL=(, ,99),UNIT=(&LOGT, ,DEFER)
| //*
| //***** MESSAGE QUEUE STATEMENTS *****
| //*
| //QBLKS DD DSN=IMS.&SYS1.QBLKS,DISP=OLD
| //SHMSG DD DSN=IMS.&SYS1.SHMSG,DISP=OLD
| //LGMSG DD DSN=IMS.&SYS1.LGMSG,DISP=OLD
| //QBLKSL DD DSN=IMS.&SYS1.QBLKSL,DISP=OLD
| //SHMSGSL DD DSN=IMS.&SYS1.SHMSGSL,DISP=OLD
| //LGMSGSL DD DSN=IMS.&SYS1.LGMSGSL,DISP=OLD
| //*
| //***** ONLINE CHANGE STATEMENTS *****
| //*
| //IMSACBA DD DSN=IMS.&SYS2.ACBLIBA,DISP=SHR
| //IMSACBB DD DSN=IMS.&SYS2.ACBLIBB,DISP=SHR
| //MODBLKSA DD DSN=IMS.&SYS2.MODBLKSA,DISP=SHR
| //MODBLKSB DD DSN=IMS.&SYS2.MODBLKSB,DISP=SHR

```



```

|          //MODSTAT DD DSN=IMS.&SYS.MODSTAT,DISP=SHR
|          //MODSTAT2 DD DSN=IMS.&SYS.MODSTAT2,DISP=SHR
|          //*
|          //***** FAST PATH STATEMENTS *****
|          //*
|          //MSDBCP1 DD DSN=IMS.&SYS.MSDBCP1,DISP=SHR
|          //MSDBCP2 DD DSN=IMS.&SYS.MSDBCP2,DISP=SHR
|          //MSDBCP3 DD DSN=IMS.&SYS.MSDBCP3,DISP=SHR
|          //MSDBCP4 DD DSN=IMS.&SYS.MSDBCP4,DISP=SHR
|          //MSDBDUMP DD DSN=IMS.&SYS1.MSDBDUMP,DISP=SHR
|          //MSDBINIT DD DSN=IMS.&SYS.MSDBINIT,DISP=SHR
|          //*
|          //***** MFS STATEMENTS *****
|          //*
|          //FORMATA DD DSN=IMS.&SYS2.FORMAT,DISP=SHR
|          //FORMATB DD DSN=IMS.&SYS2.FORMATB,DISP=SHR
|          //IMSTFMTA DD DSN=IMS.&SYS2.TFORMAT,DISP=SHR
|          //          DD DSN=IMS.&SYS2.FORMAT,DISP=SHR
|          //IMSTFMTB DD DSN=IMS.&SYS2.TFORMAT,DISP=SHR
|          //          DD DSN=IMS.&SYS2.FORMATB,DISP=SHR
|          //*
|          //***** SYSTEM STATEMENTS *****
|          //*
|          //SYSUDUMP DD SYSOUT=&SOUT,
|          //          DCB=(LRECL=125,RECFM=FBA,BLKSIZE=3129),
|          //          SPACE=(6050,300,,ROUND)
|          //IMSRDS DD DSN=IMS.&SYS.RDS,DISP=SHR
|          //IMSRDS2 DD DSN=IMS.&SYS.RDS2,DISP=SHR
|          //MATRIXA DD DSN=IMS.&SYS2.MATRIXA,DISP=SHR
|          //MATRIXB DD DSN=IMS.&SYS2.MATRIXB,DISP=SHR
|          //DFSTCF DD DSN=IMS.TCFSLIB,DISP=SHR
|          //PRINTDD DD SYSOUT=&SOUT
|          //*
|          //***** TELEPROCESSING LINE STATEMENTS *****
|          //***** GENERATED FROM THE DB/DC DEFINITION *****
|          //*
|          //* *** THE GENERATED TELETOCESSING LINE STATEMENTS WOULD BE HERE ***
|          //*
|          //* W A R N I N G: IF THIS IMS PROCEDURE WILL BE
|          //* USED FOR THE SECOND SYSTEM IN A XRF
|          //* ENVIRONMENT, THE PRIMARY MASTER TERMINAL WILL BE
|          //* DROPPED.
|          //*
|          //*
|          //*
|          //***** EXTERNAL SUBSYSTEM STATEMENTS *****
|          //*
|          //* USER MAY OPTIONALLY ADD THE DFSESL DD CARD
|          //* FOR EXTERNAL SUBSYSTEM CONNECTION.
|          //*
|          //***** DATA BASE STATEMENTS *****
|          //*
|          //* USER MAY OPTIONALLY SUPPLY THE DD STATEMENTS
|          //* FOR THE ON-LINE DATA BASES TO BE
|          //* INSERTED HERE PRIOR TO ATTEMPTING
|          //* AN ON-LINE SYSTEM EXECUTION USING
|          //* THIS PROCEDURE.
|          //* IF NO DD STATEMENTS ARE SUPPLIED FOR
|          //* A DATA BASE, IMS ASSUMES THAT THIS
|          //* DATA BASE HAS BEEN DESCRIBED THROUGH
|          //* THE DFSMDA MACRO.

```

```
|      /** IF THE USER WILL BE EXECUTING WITH THE DL/I  
|      /** SAS OPTION, THESE DD STATEMENTS SHOULD BE ADDED  
|      /** TO THE DLISAS PROCLIB MEMBER OR DESCRIBED  
|      /** THROUGH THE DFSMDA MACRO.
```

**Notes:**

- The XRF parameters are used for an XRF-capable system only.
- The RSR parameters are used for an RSR-capable system only.
- On the Monitor statements, specify `DCB=BLKSIZE=nnnnn` to ensure good performance when running the IMS Monitor. A block size should be as large as possible for your environment. A value greater than 20,000 is usually optimal. If `DCB=BLKSIZE` is not specified, the default block size is 1048 for a JCL-allocated IMSMON data set, and any data set block size value (in the DSCB) is ignored. The default blocksize value is likely to cause performance degradation for a busy IMS system.
- The IMSMON DD statement is produced only if requested in the IMSCTF macro.
- For the Fast Path statements, the DEDB areas must be allocated in the IMS control region address space, regardless of whether the DL/I SAS option is used. Allocation will be attempted in the following order:
  - The DD statements are in the control region JCL
  - DEDB dynamic allocation from the RECON data set
  - Dynamic allocation members in the IMSDALIB concatenation or JOBLIB/STEPLIB concatenation

**Recommendation:** Register the databases with DBRC and do not use DFSMDA.

## IMS Procedure Parameters

The following parameters are valid for the IMS procedure. See “Parameter Descriptions for IMS Supplied Procedures” on page 255.

	ALOT=	DSCT=	MAXPST=	RECASZ=
	AOIP=	EMHB=	MCS=	RES=
	AOIS=	EMHL=	MNPS=	RGN=
	AOI1=	EPCB= <sup>1</sup>	MNPSPW=	RGSUF=
	APPC=	ETO=	MSDB= <sup>1</sup>	RRS=
	APPCSE=	EXVR=	NHTS=	RSRMBR
	APPLID1=	FBP=	NLXB=	RVFY=
	APPLID2=	FDRMBR=	ODBASE=	SAV=
	APPLID3=	FESTIM=	ORSMBR=	SGN=
	ARC=	FIX=	OTHR= <sup>1</sup>	SHAREDQ=
	ARMRST=	FMTO=	OTMA=	SHMSGSZ=
	ASOT=	FPDSSIZE	OTMAASY=	SOD=
	AUTO=	FPOPN= <sup>1</sup>	OTMAMD=	SOUT=
	BSIZ= <sup>1</sup>	FPRLM= <sup>1</sup>	OTMANM=	SPM=
	CCTCVCAN=	FPWP=	OTMASE=	SRCH=
	CIOP=	FRE=	OTMASP=	SSM=
	CMDMCS=	GRNAME=	PASSWD1=	SUF=
	CPLOG=	GRSNAME=	PIINCR=	SVC2=
	CRC=	HIOP=	PIMAX=	SVSODR=
	CSAPSB=	HSBID=	PRDR=	SYS=
	CSLG=	HSBMBR=	PRLD=	SYS1=
	DBBF= <sup>1</sup>	MSGGROUP=	PSB=	SYS2=
	DBFP= <sup>1</sup>	MSID=	PSBW=	TCORACF=
	DBFX= <sup>1</sup>	IOVFI= <sup>1</sup>	PST=	TRACK=
	DBRCNM=	IRLM=	QBUF=	TRN=
	DBWP=	IRLMNM=	QBUFHITH=	TSR=
	DC=	ISIS=	QBUFLWTH=	UHASH= <sup>1</sup>
	DESC=	LGMSGSZ=	QBUFMAX=	UHTS=
	DLIDSIZE=	LGNR= <sup>1</sup>	QBUEFSZ=	USERVAR=
	DLINM=	LHTS=	QTL=	VAUT=
	DLIPSB=	LSO=	QTU=	VSPEC=
	DLQT=	LTERM=	RCF=	WADS=
	DMB=	LUMC=	RCFTCB=	WKAP=
	DMHVF= <sup>1</sup>	LUMP=	RECA=	YEAR4=
	DPRTY=			

**Note:** <sup>1</sup>Fast Path parameter

## IMS Procedure DD Statements

The following parameters are valid for the IMS procedure. See “DD Statement Descriptions” on page 303.

DFSDUPPM DD	MODSTAT DD
DFSESL DD <sup>1</sup>	MODSTAT2 DD
DFSOLP <sub>nn</sub> DD	MSDBCP1 DD
DFSOLS <sub>nn</sub> DD	MSDBCP2 DD
DFSTCF DD <sup>1</sup>	MSDBCP3 DD
DFSWADS <sub>n</sub> DD	MSDBCP4 DD
FORMATA DD	MSDBDUMP DD
FORMATB DD	MSDBINIT DD
IMSACBA DD	PRINTDD DD
IMSACBB DD	PROCLIB DD
IMSMON DD	QBLKS DD
IMSRDS DD	QBLKSL DD
IMSRDS2 DD	RECON1 DD <sup>1</sup>
IMSTFMTA DD	RECON2 DD <sup>1</sup>
IMSTFMTA DD	RECON3 DD <sup>1</sup>
IMSTFMTB DD	SHMSG DD
LGMSG DD	SHMSG L DD
LGMSG L DD	STEPLIB DD
MATRIXA DD	SYSTPRT DD
MATRIXB DD	SYSUDUMP DD
MODBLKSA DD	
MODBLKSB DD	

**Note:** <sup>1</sup> Optional DD statement

In addition to the following DD statements, add statements for data sets representing IMS databases that are not to be dynamically allocated.

---

## IMSBATCH Procedure

| The IMSBATCH procedure executes an IMS online batch message processing  
| address space.

### | IMSBATCH Procedure Description

The procedure shown in Figure 24 on page 243 executes an DB/DC or DBCTL online batch message processing region:

```

|          //      PROC MBR=TEMPNAME,PSB=,IN=,OUT=,
|          //      OPT=N,SPIE=0,TEST=0,DIRCA=000,
|          //      PRLD=,STIMER=,CKPTID=,PARDLI=,
|          //      CPUTIME=,NBA=,OBA=,IMSID=,AGN=,
|          //      SSM=,PREINIT=,RGN=56K,SOUT=A,
|          //      SYS2=,ALTID=,APARM=,LOCKMAX=
|          // *
|          //G      EXEC PGM=DFSRR00,REGION=&RGN,
|          //      PARM=(BMP,&MBR,&PSB,&IN,&OUT,
|          //      &OPT&SPIE&TEST&DIRCA,&PRLD,
|          //      &STIMER,&CKPTID,&PARDLI,&CPUTIME,
|          //      &NBA,&OBA,&IMSID,&AGN,&SSM,
|          //      &PREINIT,&ALTID,
|          //      '&APARM',&LOCKMAX)
|          //STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
|          //      DD DSN=IMS.&SYS2.PGMLIB,DISP=SHR
|          //PROCLIB DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
|          //SYSUDUMP DD SYSOUT=&SOUT,
|          //      DCB=(LRECL=121,RECFM=VBA,BLKSIZE=3129),
|          //      SPACE=(125,(2500,100),RLSE,,ROUND)

```

Figure 24. Procedure to Execute DB/DC or DBCTL Online Batch Message Processing Region

**Note:**<sup>1</sup> Parameters in parentheses are positional.

## IMSBATCH Procedure Parameters

The following parameters are valid for the IMSBATCH procedure. See “Parameter Descriptions for IMS Supplied Procedures” on page 255.

AGN=	OUT=
ALTID=	PARDLI=
APARM=	PREINIT=
CKPTID=	PRLD=
CPUTIME=	PSB=
DIRCA=	RGN=
IMSID=	SOUT=
IN=	SPIE=
LOCKMAX=	SSM=
MBR=	STIMER=
OBA=	SYS2=
OPT=	TEST=

## IMSBATCH Procedure DD Statements

The following DD statements are valid for the IMSBATCH procedure. See “DD Statement Descriptions” on page 303.

DFSCTL DD <sup>1</sup>	IMSLOGR DD <sup>2</sup>
DFSESL DD <sup>1</sup>	PROCLIB DD
DFSHALDB DD <sup>1</sup>	STEPLIB DD
DFSSTAT DD <sup>1</sup>	SYSHALDB DD <sup>1</sup>
FPTRACE DD <sup>1</sup>	SYSUDUMP DD
IMS DD <sup>1</sup>	

**Note:**<sup>1</sup> This statement is optional.

<sup>2</sup> If the checkpoint records required to restart the BMP do not exist in the online log data sets (OLDSn), an //IMSLOGR DD statement specifying a data set containing the checkpoint log records must be added to the BMP job's JCL.

---

## IMSCOBGO Procedure

The IMSCOBGO procedure is a three-step compile, bind, and go procedure combining the IMSCOBOL procedure with an execution step for a stand-alone DL/I batch address space.

## IMSCOBGO Procedure Description

The following sample demonstrates the IMSCOBGO procedure. The procedure assumes that:

- You supply source data from SYSIN.
- Output class is A.
- MBR=NAME, where NAME is the load module name for the program.
- SYSDA is a generic device name.

If VSAM databases are used, see "Specifying IMS Buffer Pools" on page 430.

```

//      PROC MBR=TEMPNAME,PAGES=60,SYS2=,
//      SOUT=A,RGN=4M,LOGT=2400,
//      PSB=,BUF=7,SPIE=0,TEST=0,EXCPVR=0,
//      RST=0,PRLD=,SRCH=0,CKPTID=,MON=N,
//      LOGA=0,FMT0=T,IMSID=,SWAP=,DBRC=,IRLM=,
//      IRLMNM=,BKO=N,IOB=,SSM=,APARM=,
//      LOCKMAX=,IMSPLEX=
//C      EXEC PGM=IKFCBL00,REGION=4M,
//      PARM='SIZE=130K,BUF=10K,LINECNT=50'
//SYSLIN DD DSN=&&LIN,DISP=(MOD,PASS),UNIT=SYSDA,
//      DCB=(IMS.&SYS2.PROCLIB),
//      SPACE=(3520,(40,10),RLSE,,ROUND)
//SYSPRINT DD SYSOUT=&SOUT,
//      DCB=(LRECL=121,BLKSIZE=605,RECFM=FBA),
//      SPACE=(605,(&PAGES.0,&PAGES),RLSE,,ROUND)
//SYSUT1 DD UNIT=SYSDA,DISP=(,DELETE),
//      SPACE=(3520,(100,10),RLSE,,ROUND)
//SYSUT2 DD UNIT=SYSDA,DISP=(,DELETE),
//      SPACE=(3520,(100,10),RLSE,,ROUND)
//SYSUT3 DD UNIT=SYSDA,DISP=(,DELETE),
//      SPACE=(3520,(100,10),RLSE,,ROUND)
//SYSUT4 DD UNIT=SYSDA,DISP=(,DELETE),
//      SPACE=(3520,(100,10),RLSE,,ROUND)
//L      EXEC PGM=IEWL,REGION=4M,
//      PARM='XREF,LET,LIST',
//      COND=(4,LT,C)
//SYSLIB DD DSN=SYS1.COBLIB,DISP=SHR
//SDFSRESL DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//SYSLIN DD DSN=&&LIN,DISP=(OLD,DELETE),
//      VOL=REF=*.C.SYSLIN
//      DD DISP=SHR,
//      DSN=IMS.&SYS2.PROCLIB(CBLTDLI)
//      DD DDNAME=SYSIN
//SYSLMOD DD DISP=SHR,
//      DSN=IMS.&SYS2.PGMLIB(&MBR)
//SYSPRINT DD SYSOUT=&SOUT,
//      DCB=(RECFM=FBA,LRECL=121,BLKSIZE=605),
//      SPACE=(605,(&PAGES.0,&PAGES),RLSE,,ROUND)
//SYSUT1 DD UNIT=(SYSDA,SEP=(SYSLMOD,SYSLIN)),

```

```

|          //          DISP=(,DELETE),
|          //          SPACE=(3520,(100,10),RLSE,,ROUND)
|          //G        EXEC PGM=DFSRRRC00,REGION=&RGN,TIME=2,
|          //          COND=(4,LT),
|          //          PARM=(DLI,&MBR,&PSB,&BUF,
|          //          &SPIE&TEST&EXCPVR&RST,
|          //          &PRLD,&SRCH,&CKPTID,&MON,&LOGA,
|          //          &FMTO,&IMSID,&SWAP,
|          //          &DBRC,&IRLM,&IRLMNM,&BKO,&IOB
|          //          &SSM,'&APARM',&LOCKMAX,,&IMSPLEX)
|          //STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
|          //          DD DSN=IMS.&SYS2.PGMLIB,DISP=SHR
|          //SDFSRESL DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
|          //IMS      DD DSN=IMS.&SYS2.PSBLIB,DISP=SHR
|          //          DD DSN=IMS.&SYS2.DBDLIB,DISP=SHR
|          //PROCLIB DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
|          //IEFRDER DD DSN=IMSLOG,DISP=(,KEEP),VOL=(,,99),
|          //          UNIT=(&LOGT,,DEFER),
|          //          DCB=(RECFM=VB,BLKSIZE=4096,
|          //          LRECL=4092,BUFNO=2)
|          //IEFRDER2 DD DSN=IMSLOG2,DISP=(,KEEP),VOL=(,,99),
|          //          UNIT=(&LOGT,,DEFER,SEP=IEFRDER),
|          //          DCB=(RECFM=VB,BLKSIZE=4096,
|          //          LRECL=4092,BUFNO=2)
|          //SYSOUT  DD SYSOUT=&SOUT,SPACE=(CYL,(1,1)),
|          //          DCB=(LRECL=133,RECFM=FBA,BLKSIZE=665)
|          //SYSUDUMP DD SYSOUT=&SOUT,
|          //          DCB=(LRECL=121,RECFM=FBA,BLKSIZE=3025),
|          //          SPACE=(3025,(200,100),RLSE,,ROUND)

```

### Notes

1. Parameters in parentheses are positional.
2. IMS.PROCLIB should have the same block size as SYS1.PROCLIB, which should be 80, 400, or 3200.
3. The IEFRDER and IEFRDER2 statements are not required if the job does not declare database update intent.

For a job step declaring database-update intent, DD DUMMY can be specified if the job step is not using DBRC. This is valid where an image copy of the database is taken before the update job step.

Log initialization calculates the smallest value necessary for logical record length. If the JCL logical record length value is larger than the calculated value, the JCL value is used; otherwise, log initialization uses the calculated value for logical record length and adds 4 for the block size.

If multiple volumes are required for the system log, a volume count value should be specified in the VOL parameter of the DD statement.

If the IBM 3480 tape drive is used for the IMS log data set, IMS forces tape write mode (DCB=OPCD=W). The default on the 3480 is to hold the write in a buffer so that IMS cannot detect when the write is actually performed. If a power failure occurs after a log record is written to the 3480, and the database is updated but the log record is not yet written to tape, database integrity is lost. Tape write mode is forced for the log in batch and GSAM data sets.

## IMSCOBGO Procedure Parameters

The following parameters are valid for the IMSCOBGO procedure. See “Parameter Descriptions for IMS Supplied Procedures” on page 255.

BKO=	MBR=
BUF=	MON=
CKPTID=	PRLD=
DBRC=	PSB=
EXCPVR=	RGN=
FMTO=	SOUT=
IMSID=	SPIE=
IN=	SRCH=
IMSPLEX=	SWAP=
IRLM=	SYS2=
IRLMNM=	TEST=
JOB=	
LOGA=	
LOGT=	

## IMSCOBGO Procedure DD Statements

The following DD statements are valid for the IMSCOBGO procedure. See “DD Statement Descriptions” on page 303.

In addition to the following DD statements, add statements for data sets representing IMS databases.

DFSHALDB <sup>1</sup>	SYSHALDB DD <sup>1</sup>
DFSSTAT DD	SYSLIB DD
IEFRDER DD	SYSLIN DD
IEFRDER2 DD	SYSMOD DD
IMS DD	SYSOUT DD
PROCLIB DD	SYSPRINT DD
RECON <sub>n</sub> DD	SYSUDUMP DD
SDFSRESL DD	SYSUT <sub>n</sub> DD
STEPLIB DD	

---

## IMSCOBOL Procedure

| The IMSCOBOL procedure is a two-step compile and bind procedure for IMS  
| applications written in COBOL.

### IMSCOBOL Procedure Description

| Figure 25 on page 247 demonstrates the IMSCOBOL procedure. The procedure  
| assumes that:

- You supply source data from SYSIN.
- Output class is A.
- MBR=NAME, where NAME is the load module name for the program.
- SYSDA is a generic device name.



```

//      PROC MBR=TEMPNAME,PAGES=60,SYS2=,
//      SOUT=A
//C      EXEC PGM=IKFCBL00,REGION=4M,
//      PARM='SIZE=130K,BUF=10K,LINECNT=50'
//SYSLIN DD DSN=&&LIN,DISP=(MOD,PASS),UNIT=SYSDA,
//      DCB=(IMS.&SYS2.PROCLIB),
//      SPACE=(3520,(40,10),RLSE,,ROUND)
//SYSPRINT DD SYSOUT=&SOUT,
//      DCB=(LRECL=121,BLKSIZE=605,RECFM=FBA),
//      SPACE=(605,(&PAGES.0,&PAGES),RLSE,,ROUND)
//SYSUT1 DD UNIT=SYSDA,DISP=(,DELETE),
//      SPACE=(3520,(100,10),RLSE,,ROUND)
//SYSUT2 DD UNIT=SYSDA,DISP=(,DELETE),
//      SPACE=(3520,(100,10),RLSE,,ROUND)
//SYSUT3 DD UNIT=SYSDA,DISP=(,DELETE),
//      SPACE=(3520,(100,10),RLSE,,ROUND)
//SYSUT4 DD UNIT=SYSDA,DISP=(,DELETE),
//      SPACE=(3520,(100,10),RLSE,,ROUND)
//L      EXEC PGM=IEWL,REGION=4M,
//      PARM='XREF,LET,LIST',
//      COND=(4,LT,C)
//SYSLIB DD DSN=SYS1.COBLIB,DISP=SHR
//SDFSRESL DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//SYSLIN DD DSN=&&LIN,DISP=(OLD,DELETE),
//      VOL=REF=*.C.SYSLIN
//      DD DISP=SHR,
//      DSN=IMS.&SYS2.PROCLIB(CBLTDLI)
//      DD DDNAME=SYSIN
//SYSLMOD DD DISP=SHR,
//      DSN=IMS.&SYS2.PGMLIB(&MBR)
//SYSPRINT DD SYSOUT=&SOUT,
//      DCB=(RECFM=FBA,LRECL=121,BLKSIZE=605),
//      SPACE=(605,(&PAGES.0,&PAGES),RLSE,,ROUND)
//SYSUT1 DD UNIT=(SYSDA,SEP=(SYSLMOD,SYSLIN)),
//      DISP=(,DELETE),
//      SPACE=(3520,(100,10),RLSE,,ROUND)

```

Figure 25. Compile and Bind Procedure for IMS COBOL

## IMSCOBOL Procedure Parameters

The following parameters are valid for the IMSCOBOL procedure. See “Parameter Descriptions for IMS Supplied Procedures” on page 255.

MBR=	SYS2=
SOUT=	

## IMSCOBOL Procedure DD Statements

The following DD statements are valid for the IMSCOBOL procedure. See “DD Statement Descriptions” on page 303.

SDFSRESL DD	SYSLMOD DD
SYSLIB DD	SYSPRINT DD
SYSLIN DD	SYSUTn DD

## IMSFP Procedure

The IMSFP procedure is an online procedure that initiates a Fast Path region where Fast Path applications run.

Nonmessage-driven Fast Path applications are not supported and should be changed to run as BMPs, using the IMSBATCH procedure. See “IMSBATCH Procedure” on page 242.

## IMSFP Procedure Description

The procedure shown in Figure 26 executes a Fast Path region where Fast Path applications run.

```
//IFPROC PROC MBR=TEMPNAME,SOUT=A,RGN=100K,OPT=N,
//          PSB=,NBA=000,OBA=000,TLIM=1,
//          DIRCA=000,PRLD=,STIMER=0,SOD=,DBLDL=,
//          CPUTIME=,IMSID=,AGN=,SSM=,PREINIT=,
//          SYS2=,ALTID=,APARM=,LOCKMAX=
//IFP EXEC PGM=DFSRR00,REGION=&RGN,
//        PARM=(IFP,&MBR,&PSB,&NBA,&OBA,&OPT,
//              &TLIM,&DIRCA,&PRLD,&STIMER,&SOD,&DBLDL,
//              &CPUTIME,&IMSID,&AGN,&SSM,&PREINIT,
//              &ALTID,'&APARM',&LOCKMAX)
//STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//         DD DSN=IMS.&SYS2.PGMLIB,DISP=SHR
//PROCLIB DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
//SYSUDUMP DD SYSOUT=&SOUT,
//         DCB=(LRECL=121,RECFM=VBA,BLKSIZE=3129),
//         SPACE=(125,(2500,100),RLSE,,ROUND)
```

Figure 26. Procedure to execute a Fast Path application program

**Note:** The IFP,&MBR,&PSB,&NBA,&OBA,&OPT parameters are positional.

## IMSFP Procedure Parameters

The following parameters are valid for the IMSFP procedure. See “Parameter Descriptions for IMS Supplied Procedures” on page 255.

AGN=	OPT=
ALOT=	PREINIT=
ALTID=	PRLD=
APARM=	PSB=
CPUTIME=	RGN=
DBLDL=	SOD=
DIRCA=	SOUT=
IMSID=	SSM=
LOCKMAX=	STIMER=
MBR=	SYS2=
NBA=	TLIM=
OBA=	

## IMSFP Procedure DD Statements

The following DD statements are valid for the IMSFP procedure. See “DD Statement Descriptions” on page 303.

PROCLIB DD  
STEPLIB DD

SYSUDUMP DD

## IMSMSG Job

The IMSMSG job is used to execute an IMS message processing program.

## IMSMSG Job Description

The following job executes procedure DFSMPR, which is used to start an online message region. See “DFSMPR Procedure” on page 219 for details.

```
//MESSAGE JOB 1,IMS,MSGLEVEL=1,PRTY=11,CLASS=K,MSGCLASS=A,REGION=56K
//      EXEC DFSMPR,
//      IMSID=
```

## IMSPLI Procedure

The IMSPLI procedure is a two-step compile and bind procedure for IMS applications written in PL/I.

## IMSPLI Procedure Description

The following sample demonstrates the IMSPLI procedure. The sample procedure assumes that:

- Your supply source data is from SYSIN.
- Output class is A.
- MBR=NAME, where NAME is the load module name for the program.
- SYSDA is a generic device name.

```
//      PROC MBR=TEMPNAME,PAGES=50,SYSDA=,
//      SOUT=A
//C      EXEC PGM=IELOAA,REGION=4M,
//      PARM=(XREF,A,OBJ,NODECK,NOMACRO,,
//      'OPT(TIME)')
//SYSUT1 DD UNIT=SYSDA,
//      SPACE=(1024,(60,60),RLSE,,ROUND),
//      DCB=BLKSIZE=1024,DISP=(,DELETE)
//SYSPRINT DD SYSOUT=&SOUT,
//      DCB=(LRECL=125,BLKSIZE=629,RECFM=VBA),
//      SPACE=(605,(&PAGES.0,&PAGES),RLSE)
//SYSLIN DD UNIT=SYSDA,SPACE=(80,(250,80),RLSE),
//      DCB=(IMS.&SYS2.PROCLIB),
//      DISP=(,PASS)
//L      EXEC PGM=IEWL,PARM='XREF,LIST,LET',
//      COND=(4,LT,C),REGION=4M
//SYSLIB DD DSN=SYS1.PLIBASE,DISP=SHR
//SDFSRESL DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
//SYSLIN DD DSN=*.C.SYSLIN,DISP=(OLD,DELETE)
//      DD DISP=SHR,
//      DSN=IMS.&SYS2.PROCLIB(PLITDLI)
//      DD DDNAME=SYSIN
//SYSLMOD DD DISP=SHR,
//      DSN=IMS.&SYS2.PGMLIB(&MBR)
//SYSPRINT DD SYSOUT=&SOUT,
//      DCB=(LRECL=121,RECFM=FBA,BLKSIZE=605),
//      SPACE=(605,(&PAGES.0,&PAGES),RLSE)
//SYSUT1 DD UNIT=SYSDA,DISP=(,DELETE),
//      SPACE=(CYL,(5,1),RLSE)
```



```

| //SDFSRESL DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
| //SYSLIN DD DSN=*.C.SYSLIN,DISP=(OLD,DELETE)
| // DD DISP=SHR,
| // DSN=IMS.&SYS2.PROCLIB(PLITDLI)
| // DD DDNAME=SYSIN
| //SYSLMOD DD DISP=SHR,
| // DSN=IMS.&SYS2.PGMLIB(&MBR)
| //SYSPRINT DD SYSOUT=&SOUT,
| // DCB=(LRECL=121,RECFM=FBA,BLKSIZE=605),
| // SPACE=(605,(&PAGES.0,&PAGES),RLSE)
| //SYSUT1 DD UNIT=SYSDA,DISP=(,DELETE),
| // SPACE=(CYL,(5,1),RLSE)
| //G EXEC PGM=DFSRR00,REGION=&RGN,TIME=5,
| // COND=(4,LT),
| // PARM=(DLI,&MBR,&PSB,&BUF,
| // &SPIE&TEST&EXCPVR&RST,
| // &PRLD,&SRCH,&CKPTID,&MON,&LOGA,
| // &FMTO,&IMSID,&SWAP,&DBRC,&IRLM,
| // &IRLMNM,&BKO,&IOB,&SSM,'&APARM',
| // &LOCKMAX,,&IMSPLEX)
| //STEPLIB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
| // DD DSN=IMS.&SYS2.PGMLIB,DISP=SHR
| //DFSRESLB DD DSN=IMS.&SYS2.SDFSRESL,DISP=SHR
| //IMS DD DSN=IMS.&SYS2.PSBLIB,DISP=SHR
| // DD DSN=IMS.&SYS2.DBDLIB,DISP=SHR
| //PROCLIB DD DSN=IMS.&SYS2.PROCLIB,DISP=SHR
| //IEFRDER DD DSN=IMSLOG,DISP=(NEW,KEEP),
| // VOL=(, ,99),UNIT=(&LOGT, ,DEFER),
| // DCB=(RECFM=VB,BLKSIZE=4096,
| // LRECL=4092,BUFNO=2)
| //IEFRDER2 DD DSN=IMSLOG2,DISP=(NEW,KEEP),
| // UNIT=(&LOGT, ,DEFER,SEP=IEFRDER),
| // VOL=(, ,99),DCB=(RECFM=VB,BLKSIZE=4096,
| // LRECL=4092,BUFNO=2)
| //SYSPRINT DD SYSOUT=&SOUT,
| // DCB=(LRECL=121,BLKSIZE=605,RECFM=FBA),
| // SPACE=(605,(500,500),RLSE, ,ROUND)
| //SYSUDUMP DD SYSOUT=&SOUT,
| // DCB=(LRECL=121,BLKSIZE=605,RECFM=FBA),
| // SPACE=(605,(500,500),RLSE, ,ROUND)

```

**Notes:**

1. These parameters are positional.
2. IMS.PROCLIB should have the same block size as SYS1.PROCLIB, which should be 80, 400, or 3200.
3. The IEFRDER and IEFRDER2 statements are not required if the job does not declare database-update intent.

For a job step declaring database-update intent, DD DUMMY can be specified if the job step is not using DBRC. This is valid where an image copy of the database is taken before the update job step.

The log-initialization process calculates the smallest value necessary for logical record length. If the JCL logical record length value is larger than the calculated value, the JCL value is used; otherwise, the log-initialization process uses the calculated value for logical record length and adds 4 for the block size.

If multiple volumes are required for the system log, a volume count value should be specified in the VOL parameter of the DD statement.

If the IBM 3480 tape drive is used for the IMS log data set, IMS forces tape write mode (DCB=OPCD=W). The default on the 3480 is to hold the write in a buffer so that IMS cannot detect when the write is actually performed. If a power failure occurs after a log record is written to the 3480, and the database is updated but the log record is not yet written to tape, database integrity is lost. Tape write mode is forced for the log in batch and GSAM data sets.

## IMSPLIGO Procedure Parameters

The following parameters are valid for the IMSPLIGO procedure. See “Parameter Descriptions for IMS Supplied Procedures” on page 255.

BKO=	MBR=
BUF=	MON=
CKPTID=	PRLD=
DBRC=	PSB=
EXCPVR=	RGN=
FMTO=	RST=
IMSID=	SOUT=
IMSPLEX=	SPIE=
IOB=	SRCH=
IRLM=	SWAP=
IRLMNM=	SYS2=
LOGA=	TEST=
LOGT=	

## IMSPLIGO Procedure DD Statements

The following DD statements are valid for the IMSPLIGO procedure. See “DD Statement Descriptions” on page 303.

DFSHALDB <sup>1</sup>	SYSHALDB DD <sup>1</sup>
DFSRESLB DD	SYSLIB DD
DFSSTAT DD	SYSLIN DD
IEFRDER DD	SYSLMOD DD
IEFRDER2 DD	SYSOUT DD
IMS DD	SYSPRINT DD
PROCLIB DD	SYSUDUMP DD
RECON <sub>n</sub> DD	SYSUT1 DD
SDFSRESL DD	

---

## IMSRDR Procedure

| The IMSRDR procedure is a procedure used to read an IMSMSG job into the  
| operating system job stream from direct access storage devices (DASD).

## IMSRDR Procedure Description

The DASD procedure shown in Figure 27 on page 253 reads an IMSMSG job into the operating system job stream from direct access storage devices.

```

|          //          PROC MBR=IMSMMSG,CLASS=A,SYS2=
|          //IEFPROC EXEC PGM=IEBEDIT
|          //SYSPRINT DD DUMMY
|          //SYSUT1   DD DDNAME=IEFRDER
|          //SYSUT2   DD SYSOUT=(&CLASS,INTRDR),DCB=BLKSIZE=80
|          //SYSIN    DD DUMMY
|          //IEFRDER  DD DISP=SHR,
|          //          DSN=IMS.&SYS2.JOBS(&MBR)

```

Figure 27. DASD Procedure to Read IMSMSG Job

## IMSRDR Procedure Parameters

The following parameters are valid for the IMSRDR procedure. See “Parameter Descriptions for IMS Supplied Procedures” on page 255.

```

CLASS=          SYS2=
MBR=

```

## IMSRDR Procedure DD Statements

The following DD statements are valid for the IMSRDR procedure. See “DD Statement Descriptions” on page 303.

```

IEFRDER DD          SYSUT1 DD
SYSIN DD           SYSUT2 DD
SYSPRINT DD
EXCPVR=
FMTO=

```

---

## RDIBATCH Procedure

The RDIBATCH procedure is used to help maintain database availability in failure situations.

The RDI region is used in a block-level data sharing environment to provide a path from the IRLM to DBRC to guarantee database availability if a failure occurs. The RDI region accomplishes the following:

- IDENTIFY to the IRLM
- SIGNON to DBRC
- WAIT for an IRLM failure.

If an IRLM fails, its associated RDI region abends with a U3303. The surviving IRLM partner is notified of the failure and, in turn, notifies DBRC of the failure.

The RDI region can only be terminated by an IRLM failure or by an operator CANCEL command. You must ensure that the IRLM and DBRC are both being used to gain the benefits of data sharing protection that this region offers.

The procedure shown in Figure 28 on page 254 is **not** placed in IMS.PROCLIB by IMS system definition. You must create it by modifying a copy of either the DBBBATCH or DLIBATCH procedures. The region type is RDI.

**Related Reading:** If you are using IMS with CICS in an XRF environment, this procedure is essential. For more information about using IMS with CICS in and XRF environment, see *CICS Extended Recovery Facility Guide*.

## RDIBATCH Procedure Description

The sample procedure shown in Figure 28 demonstrates the RDIBATCH procedure.

```
//      PROC MBR=ANYNAME, 1 PSB=, BUF=,
//      SPIE=0, TEST=0, EXCPVR=0, RST=0, PRLD=,
//      SRCH=0, CKPTID=, MON=, LOGA=0, FMTO=T,
//      IMSID=, SWAP=, DBRC=Y, 2 IRLM=Y, 3
//      IRLMNM=IRLM, 4 BKO=N, IOB=, SSM=, APARAM=,
//      RGN=2048K,
//      SOUT=A,
//      SYS=,
//      SYS2=, LOCKMAX=, IMSPLEX=
//G     EXEC PGM=DFSRRCO0, REGION=&RGN, TIME=1440, 5
//      PARM=(RDI, &MBR, &PSB, &BUF,
//      &SPIE&TEST&EXCPVR&RST, &PRLD,
//      &SRCH, &CKPTID, &MON, &LOGA, &FMTO,
//      &IMSID, &SWAP, &DBRC, &IRLM, &IRLMNM,
//      &BKO, &IOB, &SSM,
//      '&APARM', &LOCKMAX, , , &IMSPLEX)
//STEPLIB DD DISP=SHR, DSN=IMS.&SYS2.SDFSRESL
//DFSRESLB DD DISP=SHR, DSN=IMS.&SYS2.SDFSRESL
//PROCLIB DD DISP=SHR, DSN=IMS.&SYS.PROCLIB
//RECON1 DD DISP=SHR, DSN=IMS.&SYS.RECON1
//RECON2 DD DISP=SHR, DSN=IMS.&SYS.RECON2
//RECON3 DD DISP=SHR, DSN=IMS.&SYS.RECON3
//SYSABEND DD SYSOUT=&SOUT.
        PEND
```

Figure 28. RDIBATCH Procedure

### Notes:

1. MBR=xxxxxxx must be specified but does not need to be valid. The specified member will not receive control but must be specified or parameter analysis fails.
2. DBRC=Y is required (specified or used by default).
3. IRLM=Y is required (specified or used by default).
4. IRLMNM=cccc is required (specified or used by default).
5. TIME=1440 is recommended.

## RDIBATCH Procedure Parameters

The following parameters are valid for the RDIBATCH procedure. See “Parameter Descriptions for IMS Supplied Procedures” on page 255.



APARM=	MBR=
BKO=	MON=
BUF=	PRLD=
CKPTID=	PSB=
DBRC=	RGN=
EXCPVR=	RST=
FMTO=	SOUT=
IMSID=	SPIE=
IMSPLEX=	SRCH=
IOB=	SSM=
IRLM=	SWAP=
IRLMNM=	SYS=
LOCKMAX=	SYS2=
LOGA=	TEST=

## RDIBATCH Procedure DD Statements

DFSRESLB DD	STEPLIB DD
PROCLIB DD	SYSABEND DD
RECON <sub>n</sub> DD	

The following DD statements are valid for the RDIBATCH procedure. See “DD Statement Descriptions” on page 303.

---

## Parameter Descriptions for IMS Supplied Procedures

This section describes all of the parameters that can be specified by the procedures supplied for IMS.

### AGN= through AUTO=

#### AGN=

Specifies the one- to eight-character application group name used for interregion communication security. The default is blanks.

**Related Reading:** For details on the use of the application group name, refer to the Security Maintenance utility information in *IMS Version 9: Administration Guide: System*.

#### AOI1=

Specifies the security product and level of security to use for authorization of type 1 AOI commands. The default value for this parameter is set during system definition.

**A** Includes options C and R. RACF is called first, and then the system authorization facility (SAF) return code, RACF return code, and RACF reason code are passed to the Command Authorization exit routine (DFSCCMD0). These return codes are decoded into a security code, which is also passed to the user exit DFSCCMD0 for processing.

**N** Specifies that no authorization security checking is to be used for this function.

**C** Specifies that DFSCCMD0 is to be called for command authorization.

**R** Specifies that RACF is to be called for command authorization.

**S** Specifies that Security Maintenance utility (SMU) security is to be called for command authorization.

| Because the AOI1 specification is not included in a checkpoint record, you can  
| change the AOI1 value each time IMS is initialized

**ALOT=**

Specifies the auto logoff time in minutes. Valid values are 0 and from 10 to 1440. If the ALOT value is not specified, the value from the JCL member is used except for FINANCE, SLU P, and ISC. If ALOT is not specified on the logon descriptor or overridden by the logon exit (DFSLGNX0) for FINANCE, SLU P, and ISC, a value of 1440 is used (the value from the JCL member is ignored).

- ALOT=0

The terminal is logged off immediately when no signon is in effect. ALOT=0 is normally specified when signon occurs by one of the following methods:

- The user is signed on automatically during the logon process (autologon)
- Signon data is supplied by logon (BIND) userdata
- Signon data is supplied by the logon exit (DFSLGNX0)

**Restriction:** Do not specify ALOT=0 for interactive terminal sessions if the session expects a response to the DFS3649 (signon required) salutation message.

- ALOT=(10-1439)

The session is terminated after the specified number of minutes have elapsed without a signed-on user.

- ALOT=1440

The session is never automatically terminated.

ALOT only applies to ETO terminals.

**ALTID=**

Specifies a one- to four-character name for an alternate IMS system. When your IMS message processing region tries to link to an IMS system, it tries to link to the system specified by the IMSID= parameter. If no IMS system matches the name specified with IMSID, the message processing region tries the system specified with the ALTID= parameter. If no IMS system matches the name specified with ALTID=, what happens next depends upon the option you choose on the OPT= parameter.

**AOIP=**

Specifies a value for the upper expansion limit of the AOI buffer pool. The value can be specified as one- to six-numeric characters, or one- to five-numeric characters followed by K (kilobyte), M (megabyte), or G (gigabyte). If K, M, or G is not specified, K is the default. The maximum value is 2G-1. If the specified value exceeds 2G-1, the default is 2G-1. If a value is not specified, the upper limit default is 2G-1.

**AOIS=**

Specifies whether RACF and the Command Authorization exit routine (DFSCCMD0) are to be used with AO applications for command security.

**Related Reading:** For more information, see “Establishing IMS Security” in the *IMS Version 9: Administration Guide: System*. The possible values are:

- A** Includes options C and R. RACF is called first. Then the security authorization facility (SAF) return code, RACF return code, and RACF reason code are passed to DFSCCMD0. These return codes are decoded into a security code, which is also passed to DFSCCMD0 for processing.

- C** Specifies that DFSCCMD0 is to be called for command authorization.
- N** Specifies that the DL/I call ICMD cannot be issued by an application program.
- R** Specifies that RACF is to be called for command authorization.
- S** Specifies that no authorization checking is to be done.

The default is N.

Because AOIS is not included in the checkpoint record, you can change the AOIS value each time IMS is initialized.

#### **APARM=**

Specifies the 1 to 32-character parameter that is passed to the application program as part of the information returned in the INQY call with the ENVIRON subfunction. The parameter must be enclosed in single-quotes (') if special characters are used. Embedded commas (,) are not allowed.

#### **APPC=**

Specifies to activate (Y) or not to activate (N) APPC/IMS LU 6.2 support. The default is (N). The /STA APPC command overrides APPC=N. An IMS APPLID override can also be specified via the existing APPLID1= execution parameter during initialization.

#### **APPCSE=**

Specifies the type of APPC RACF security. The value you specify for this keyword has an equivalent value you can specify in the /SECURE APPC command. The first character you specify applies to the APPCSE keyword. The second character you specify becomes the equivalent value for the /SECURE APPC command.

The /SECURE APPC command overrides the value you specify in the APPCSE= keyword.

- |   |         |   |
|---|---------|---|
| <b>C</b>  | CHECK   | Specifies APPC RACF command and transaction security check.   |
| <b>F</b>  | FULL    | Specifies APPC RACF command, transaction security check, and additional security for dependent regions. |
| <b>N</b>  | NONE    | Specifies no APPC RACF security within IMS. (Specifying NONE does not affect APPC/MVS RACF security.)   |
| The IMS security exits (DFSCSTRN0 and DFSCCMD0) are still called if they are defined to the system. |         |   |
| <b>P</b>  | PROFILE | Specifies the use of APPC transaction security in the TP profile or the default profile.                |

The default is F.

#### **APPLFE=**

This optional parameter specifies a 1 to 8-character name of an application front-end routine that is called whenever a message processing program (MPP) is scheduled in a message processing region (MPR).

If APPLFE= is specified, the IMS MPP controller performs the following functions:

- Loads the application front-end routine.
- Calls the application front-end routine for initialization processing.
- Calls the application front-end routine when an MPP is scheduled in an MPR.
- Calls the application front-end routine for shutdown processing when the MPP terminates.

The following describes the interface between the IMS MPP controller and the application front-end routine:

1. The IMS MPP controller calls the application front-end routine for initialization processing with the input registers set as follows:
  - R0=0
  - R1=0
  - R13=address of caller's save area
  - R14=return address of caller
  - R15=entry point address of the application front-end routineThe application front-end routine must restore all registers, except register 15, upon return to IMS. Register 15 contains the return code.
2. The IMS MPP controller calls the application front-end routine when an MPP is scheduled in an MPR with the input registers set as follows:
  - R0=entry point address of the application MPP
  - R1=PCB list to be passed to the application MPP
  - R13=address of caller's save area
  - R14=return address of caller
  - R15=entry point address of the application front-end routineThe application front-end routine must restore all registers, except register 15, upon return to IMS. Register 15 contains the return code.
3. The IMS MPP controller calls the application front-end routine to perform shutdown processing when the MPP is terminating:
  - R0=non-zero
  - R1=0
  - R13=address of caller's save area
  - R14=return address of caller
  - R15=entry point address of the application front-end routineThe application front-end routine must restore all registers, except register 15, upon return to IMS. Register 15 contains the return code.

**Note:** Because the IMS MPP controller establishes an ESTAE, a program interrupt is intercepted and treated as an application failure when it occurs after the application front-end routine is called and before it returns to the IMS MPP controller. As a result, the IMS MPP controller reinitializes itself, and once again loads and calls the application front-end routine for initialization processing. IMS stops the transaction that is responsible for the interrupt.

#### **APPLID1=name1**

Specifies the application identification for the active IMS subsystem.

The VTAM APPLID is overridden by the APPLIDn= startup parameter, and APPLID3 is always used for the tracking subsystem. You must take care not to cause the same IMS VTAM APPLID to be opened for more than one IMS subsystem. If necessary, the APPLID=(name1,name2,name3) on the COMM macro should all be overridden by using the APPLID1, APPLID2, and APPLID3 parameters to ensure unique APPLIDS.

In a generic resource group environment, the APPLID1 parameter overrides the value specified in the first position of the APPLID= field of the IMS COMM macro. Using this execution parameter allows IMS system definitions to be identical by providing a unique APPLID override when starting the IMS job.

I If MNPS is used for XRF systems, the APPLID1 and APPLID2 parameters  
 I specify the application identification for the APPLID ACB. The MNPS parameter  
 I specifies the application identification for the MNPS ACB.

**APPLID2=name2**

Specifies the application identification for the XRF alternate IMS subsystem.

**APPLID3=name3**

Specifies the application identification for the RSR tracking IMS subsystem.

*name3* can match either *name1* or *name2*, but no two application identifications can be active in the network at the same time.

**ARC=**

non-zero entry (01-99) specifies that automatic archiving is to be performed. ARC= also indicates the number of online data sets (OLDSs) to be filled before issuing the DBRC GENJCL ARCHIVE command.

A /DBR command can cause automatic archiving, even if the number specified in ARC= is not reached.

IF ARC=00 is specified, no automatic archiving is performed.

The default is 01.

**ARMRST=**

Specifies whether (Y) or not (N) IMS will register with z/OS ARM and allow ARM to restart IMS in case of a failure. The default is Y.

**ASOT=**

Specifies the autosignoff time, in minutes. Valid values are 0 and from 10 to 1440. You can override this value during signon with the user descriptor. If ASOT is not specified on either the user descriptor, the logon descriptor, or overridden by either the logon (DFSLGNX0) or signon (DFSSGNX0) exits, the value from the JCL member is used except for FINANCE, SLU P, and ISC. If ASOT is not specified on either the user or logon descriptor for FINANCE, SLU P, and ISC, a value of 1440 is used (the value from the JCL member is ignored).

- ASOT=0

The user is signed off immediately when no output is available to be sent. ASOT=0 is normally specified when no IMS input or output is available, or after the last available output message completes.

**Restriction:** Do not specify ASOT=0 for interactive terminals, for example 3270s or SLU2.

- ASOT=(10-1439)

The user is signed off after the specified number of minutes have elapsed without terminal activity, irrespective of message status.

- ASOT=1440

The user is never automatically signed off.

ASOT only applies to ETO terminals.

**AUTO=**

Specifies whether (Y) or not (N) automatic restart is to be invoked. The default is N.

**BKO= through CSLG=****BKO=**

Specifies whether (Y) or not (N) dynamic backout is to be performed; BKO=

applies only to pseudoabends. Dynamic backout is to the last sync point. Y only applies if the log data set is assigned to DASD. The default is N.

**BPECFG=**

Identifies the member in the data set identified in the PROCLIB DD statement for the Recovery Data Manager of ORS.

**BSIZ=**

Specifies the database buffer size; the value can be 512, 1024, 2048, 4096, or a multiple of 4K (4096). You must specify a value large enough to contain the largest defined DEDB control interval size. The maximum value is 28672.

**BUF=**

Specifies a one to three-digit number defining the number of 1K blocks to be used in calculating the size of the OSAM subpools. This parameter only applies to a DL/I batch environment; it is used only if the DFSVSAMP data set is not supplied or does not include any IOBF control statements.

For more information, see “OSAM Buffer Pool Compatibility Definition” on page 431.

**CCTCVCAN=**

Specifies whether (Y) or not (N) IMS converts abend code 222 TO 08E when cancelling a coordinator controller (CCTL) address space that is connected to IMS.

**Attention:** If the CCTL is registered with the automatic restart manager (ARM), converting the abend code to a 08e might result in ARM restarting the CCTL.

**N** IMS does not convert abend code 222 to 08e. When you cancel the CCTL, IMS might issue abend code 0113. This would only happen if a DLI call contains an active CCTL thread. N is the default.

**Y** IMS converts abend code 222 to 08e. This enables the database recovery adapter (DRA) to recover and also prevents IMS from issuing abend code 0113.

**CIOP=**

Specifies a value for the upper expansion limit of the communication I/O buffer pool. The value can be specified as one- to six-numeric characters, or one- to five-numeric characters followed by K (kilobyte), M (megabyte), or G (gigabyte). If K, M, or G is not specified, K is the default. The maximum value is 2G-1. If the value specified exceeds 2G-1, the default is 2G-1. If a value is not specified, the upper limit default is 2G-1.

**CKPTID=**

Specifies the checkpoint at which the program is to be restarted.

Enclose the checkpoint ID in single quotes if it contains any non-ANSI characters, such as the slash.

**Related Reading:** See “XRST Call” in *IMS Version 9: Application Programming: Database Manager* for more information on specifying values for CKPTID.

You can suppress message DFS681I or DFS0540I each time a checkpoint call is made:

- To suppress message DFS681I, code NOMSG681.
- To suppress message DFS0540I, code NOMSG540.
- To suppress both messages, code NOMSGS.

The values NOMSG681, NOMSG540, and NOMSGS are not valid checkpoint restart IDs.

**CL1=,CL2=,CL3=,CL4=**

Is a required positional parameter that specifies four three-digit decimal numbers indicating which classes of messages are to be processed by this message region. For example, if classes 1, 2, and 3 are to be processed by this region, the PARM field must be specified as PARM='MSG,001002003000'. The maximum value for a class is 999.

The sequence of specifying the classes determines relative class priority within the message region. In the example, all class 1 messages are selected for scheduling before any class 2 messages are considered. Class numbers cannot be greater than the maximum number of classes specified during system definition.

**CLASS=**

Specifies the default message class for the job being submitted to the internal reader.

**CMDMCS=**

Table 28 shows the procedures that describe the valid keywords for each online system type.

*Table 28. Procedures for Online Systems*

Online System Type	Procedure Name
DB/DC	IMS
DCCTL	DCC
DBCTL	DBC

The following sub-parameters apply to the IMS and DCC procedures.

- N** Commands cannot be entered from an MCS console. N is the default.
- Y** Commands can be entered from an MCS or E-MSC console by entering the command recognition character (CRC) followed by the command text.
- R** Commands can be entered from an MCS console in the form CRC followed by the command text. The system calls RACF (or equivalent) to verify that the user ID of the console is authorized to issue the command.
- C** Commands can be entered from an MCS console in the form CRC followed by the command text. DFSCCMD0 is called to verify that the user ID of the console is authorized to issue the command.
- B** Include options R and C. RACF (or equivalent) is called first. After RACF is called, the SAF return code, RACF return code and RACF reason code are passed to DFSCCMD0. The return codes are decoded into a security code that is also passed to DFSCCMD0. DFSCCMD0 determines if the command is authorized for the user ID of the console.

This is an optional sub-parameter. If you do not select B, C, or R, commands can be entered from an MCS console. Neither RACF nor DFSCCMD0 is called to verify that the user ID of the console is authorized to issue the command.

The following sub-parameters apply to the DBC procedure.

- R** RACF (or equivalent) is called to verify that the user ID of the console is authorized to issue the command.

- C** DFSCCMD0 is called to verify that the user ID of the console is authorized to issue the command.
- B** Include options R and C. RACF (or equivalent) is called first. After RACF is called, the SAF return code, RACF return code and RACF reason code are passed to DFSCCMD0. These return codes are decoded into a security code that is also passed to DFSCCMD0. DFSCCMD0 determines whether the command is authorized for the user ID of the console.

**Note:** CRC for DB/DC does not work for restart commands. The ability to issue IMS commands using the CRC or IMSID in a DB/DC environment is available if the proper specification of the CMDMCS parameter is supplied. However, this does not work for restart commands /ERE and /NRE. You must enter these commands the same as before. After restart is complete, you can enter commands using the CRC or the IMSID.

#### **CORE=**

**Note:** This parameter is no longer used. CORE= has been replaced by the PIINCR= and PIMAX= execution parameters.

#### **CPLOG=**

Specifies the number of system log records between system-generated checkpoints. The values that are valid are between 1000 to 16 million. The default is 500000. The value specified must be one or more numeric characters followed by either a K or M (for example, 16M).

#### **CPUTIME=**

Specifies the task timing option for either BMP or Fast Path.

- 0** No task timing is performed for this BMP or Fast Path region. A default value is placed in the 07 accounting log record to indicate that no timing was done. This is the default.
- N** A one- to four-digit number specifying the maximum task time in minutes to be allowed for the execution of this BMP or Fast Path region. The values can range from 1 to 1440 minutes. If a value of 1440 or greater is specified, a time of 24 hours is used. When N is nonzero, the STIMER=1 option is in effect.

For BMP task-timing, the measured time includes DL/I processing occurring in the dependent region. When the dependent region exceeds the specified time, a U0240 abend occurs after the current DL/I call completes. To control the execution time of a BMP, use CPUTIME=N instead of the TIME= of the operating system. If the operating system time limit is included with a job that specifies CPUTIME=N, ensure that the former is a sufficiently higher value to allow for initialization and one completed DL/I call beyond the CPUTIME value. When the CPUTIME=N option is used, the application program must not use the STIMER or TTIMER CANCEL.

For Fast Path programs, the timer is set at the beginning of program invocation. However, message-driven programs are not allowed to exceed the amount of time defined for the program to process a single transaction. In this case, CPUTIME=*n* is ignored.

#### **CRC=**

CRC specifies a one-character command recognition character. Commands entered at an z/OS system console with the CRC as a prefix, are processed by the DBCTL control region that has the CRC defined.



CRC overrides the value specified on the CMDCHAR keyword of the System Definition IMSCTRL macro.

If CMDCHAR is not specified, and CRC is not specified, the default value is /.

If CMDCHAR=NONE is specified, and CRC= is not specified, commands to the DBCTL control region are prefixed by the IMSID.

The CRC must be different from any other CRC in DBCTL regions already executing. Choose an unreserved character that is appropriate for your particular system. A comma can not be specified as a command recognition character.

Do not use a character for the CRC that is the same as the beginning character of an z/OS command. If the CRC is the same as the beginning character of an z/OS command, that command does not work after you start IMS. For example, if you start IMS with CRC=D, z/OS does not respond to any of the z/OS display commands such as D A,L.

The CRC value you select might affect multisegment command processing. The last character of the command is checked during multisegment command processing. If that character matches the CRC, the command is assumed to be a multisegment command. Processing of the command waits for the rest of the command. For example, if you use B for the command recognition character, and issue the command 'BMODIFY PREPARE ACBLIB', the command does not run. The ending 'B' in 'ACBLIB' is interpreted to indicate a multisegment command. To enter this command correctly, enter:

```
'BMODIFY PREPARE ACBLIB . '
```

**Related Reading:** Refer to *IMS Version 9: Customization Guide* for more information on CRC.

### CSAPSB=

When the DL/I address space option (LSO=S) is specified, two program specification block pools are used. CSAPSB specifies the size of pool in the z/OS common storage area (CSA), and DLIPSB specifies the size in the DL/I address space. In an LSO=S system, the PSB= parameter is ignored.

The output of the ACBGEN utility indicates maximum CSA space, average CSA space, and the CSA space required by each PSB in the pool. This output is useful in determining what value to specify for the CSAPSB= parameter.

Similarly, before specifying the DLIPSB= parameter, you should consider the secondary address space (SAS) size required for each PSB, maximum SAS space, and average SAS space. Neither parameter should be zero. Normally, DLIPSB should be larger than CSAPSB.

The values for the pool sizes can be specified as either one- to six-numeric characters, or one- to five-numeric characters followed by either K (kilobyte), M (megabyte), or G (gigabyte). If K, M, or G is not specified, K is the default. The maximum value is 2G-1. The upper limit defaults to 2G-1 if any of the following statements is true:

- The value specified is not large enough to hold the largest primary and secondary storage allocations that are defined.
- The value specified exceeds 2G-1.
- No value is specified.

The sizes specified are rounded up to the nearest page boundary. The sizes of the two pools can also be specified during IMS system definition, using the SASPSB parameter in the BUFPOOLS macro.

The sum of the values for the CSAPSB and DLIPSB defines the PSB pool size. If PSB is also specified, the larger value (PSB or the sum of CSAPSB and DLIPSB) is used.

For CSAPSB and PSBW, DFSIINS0 obtains contiguous space in the z/OS common area. If this storage is unavailable in the z/OS common area, ABENDU0717 occurs.

#### **CSLG=**

Specifies a three-character suffix for the Common Service Layer IMS.PROCLIB member, DFSCGxxx. When you specify this parameter, IMS uses the Common Service Layer (CSL) to manage and operate the IMSplex. The minimum configuration for the CSL is to have one SCI and one OM on each z/OS image where an IMS control region resides.

This parameter must be explicitly specified. There are no default values.

### **DBBF= through DSCT=**

#### **DBBF=**

Specifies a one- to five-digit maximum number of buffers in ECSA for Fast Path databases. The default for this parameter is specified on the BFALLOC parameter of the FPCTRL macro during system definition; or the default of 10 is used if no value is specified. The value specified by this parameter can be increased, but not decreased, at emergency restart. The maximum value for DBBF= is 65535.

#### **DBD=**

Specifies the Fast Path Data Entry Database (DEDB) DBD name. DBD= can be from one to eight characters.

#### **DBFP=**

Specifies a one- to four-digit value that determines how often the IMS Fast Path buffer pool manager can page free unneeded buffers. The default of 0 indicates that any unneeded buffers are page freed at each dependent region termination.

Specifying a value for this parameter can improve z/OS performance when many IMS dependent regions with NBA and OBA specified are started and terminated in a short time period (for example, within a period of one second). When a large number of page-fixed buffers exist within the z/OS system, specifying a value for this parameter can decrease the z/OS overhead required to page fix and page free buffers.

This parameter alters only the method of managing the page fixing and page freeing of the IMS Fast Path buffers in the IMS Fast Path buffer pool. It does not affect the number of buffers in the buffer pool.

**0** Allows the number of page-fixed buffers to fluctuate at each termination.

**1** Allows the number of IMS Fast Path page-fixed I/O buffers to increase, without a corresponding free. If this option is selected, the number of page-fixed buffers only increases.

#### **2-3600**

Determines how often (in seconds) unneeded buffers may be page freed. During this time interval, the number of page-fixed buffers only increases; no page freeing is performed. When the specified time interval expires, the number of page-fixed buffers is reset to the minimum number of buffers needed to satisfy the current requirements.

**DBFX=**

| Specifies a one- to five-digit number of additional buffers (not separate pools) to  
 | be page-fixed at control region initialization. This specification allows for  
 | asynchronous processing, which ensures that the DEDB updates are held until  
 | the associated log buffer is written. The default is 4.

**DBLDL=**

An optional four-digit number that specifies the maximum number of BLDL entries to be kept for application modules that are not preloaded. The default maximum is 20. The maximum value is 9999.

**Related Reading:** For more information, see references to BLDL in *IMS Version 9: Administration Guide: System*.

**DBRC=**

Specifies whether database recovery control is to be used during this execution of IMS. The possible values for DBRC= are:

**C** This value has no meaning except during a batch backout run of IMS. If specified for other than a batch backout run, it is treated as a null.

With DBRC=C, you can back out batch jobs that terminate normally during an execution that includes DBRC **but not IRLM**. During batch backout of jobs that terminated normally, IMS performs authorization for databases and then backs out database changes, just as is done for jobs that terminate abnormally.

If DBRC=C is specified when the previous execution of IMS included both DBRC and IRLM, batch backout fails if the previous execution completed normally.

**N** Specifies that DBRC is not used during this execution unless DBRC=FORCE is specified on the IMSCTRL macro statement during IMS system definition. If DBRC=FORCE is specified during IMS system definition and if this is not a batch backout execution of IMS, a message is issued, and a nonzero return code is returned.

If this is a batch backout execution of IMS, the DBRC=FORCE system definition specification can be overridden by specifying DBRC=N on the EXEC procedure. Thus, if the previous execution of IMS used DBRC but not IRLM, batch backout will execute without DBRC.

**Null** Specifies whether DBRC is included, based on the specification of the IMSCTRL macro statement keyword DBRC= during IMS system definition.

**Y** Specifies that DBRC is used during this execution of IMS.

If the IRLM is not being used, DBRC provides additional database security. If a common RECON data set is shared among subsystems, DBRC allows proper authorization to registered databases so that the subsystems can share data at the database level.

**Attention:** If DBRC is used to maintain the integrity of databases, DBRC=N should be specified only when DBRC is unavailable or when you are sure that the integrity of databases will not be destroyed. When DBRC=N, before reactivating DBRC, you must register information on the log volumes created while DBRC was inactive.

**DBRCNM=**

Specifies the member in SYS1.PROCLIB containing the procedure for the DBRC address space. The IMS control region automatically starts the DBRC

procedure, using this specification. The name can be up to eight characters, the first of which must be alphabetic. The default is DBRC.

**DBRSE=**

Specifies the eight-character RSENAME. It is used by a primary and an alternate DBCTL region in a “DBCTL standby” configuration, and it must be the same in both DBCTL regions.

**Related Reading:** For more information on RSENAME, refer to *IMS Version 9: Administration Guide: System*.

**DBWP=**

Specifies the amount of subpool 231 storage to be allocated to the database work area pool. The value can be specified as either one- to six-numeric characters, or one- to five-numeric characters followed by either K (kilobyte), M (megabyte), or G (gigabyte). If K, M, or G is not specified, K is the default. The maximum value is 2G-1. If the value specified exceeds 2G-1, the default is 2G-1. The pool size specified is rounded up to the nearest page boundary.

The DBWP size must be specified as the largest logical record length parameter size of any database you might run.

**DC=**

Specifies the three-character suffix for the DFSDCxxx member in IMS.PROCLIB. The default is 000.

DFSDCxxx parameters allow the override of the VTAM node name and the LTERM names for the master and secondary terminals that are defined in the system definition.

**DEADLOK='iiii,kkkk'**

Specifies the local deadlock-detection interval in seconds or milliseconds (*iiii*) and the number of local cycles (*kkkk*) that are to occur before a global detection is initiated.

*iiii* Is a one- to four-digit number from 1 to 9999 that specifies the length of time in seconds (1 - 99) or milliseconds (100 - 9999) that will be used for the IRLM local deadlock detection interval. Values less than 100 will be recognized in seconds with 5 seconds being the maximum used. Values from 100 to 9999 are treated as milliseconds and IRLM truncates this to even 100 millisecond increments, with a maximum value of 5000 (5 seconds).

**NOTE:** Once IMS TIMEOUT candidates have timed out, they remain timeout candidates and are presented to the timeout exit each Global deadlock cycle. IMS creates SMF 79.15 records when candidates are presented which are written to the SMF data sets (if enabled). If timeout candidates are found and the value for *iiii* is subsecond, there will be many SMF 79.15 records written per second until the tasks are no longer waiting in IRLM.

*kkkk* Specifies the number of local deadlock cycles performed before global deadlock detection is performed. The default is 1.

The valid value for *iiii* is a number between 1 and 5. The value used for *kkkk* is always 1.

In data sharing environments, IRLM synchronizes all of the DEADLOK values in the group to the values specified on the most recent IRLM to join the group. The DEADLOK value may be changed by starting a member with the values desired, or by using the MODIFY irlmproc, SET, DEADLOCK= command. It is recommended that the installation specify the same value for DEADLOK on all

of its IRLM start-up procedures, and use the START irlmproc command or the MODIFY command to override this value only when the interval must be changed from its original value. If all members of a sysplex group do not support subsecond deadlock processing, the value range used by the group will be 1 to 5 seconds, with values specified less than one second, changed to one second.

**Recommendation:** Specify the same values for the DEADLOK parameter on all of your IRLM startup procedures if you don't want the value for the last IRLM to connect to the group to become the value for all.

Deadlock processing time is minimal if the number of locks that are HELD and WAITING is small. However, deadlock processing **can** affect system performance when IRLM encounters a real deadlock situation.

The amount of time to run deadlock processing is directly proportional to the number of resources HELD (this is a minor time component) plus the number of WAITERS (this is a major time component) on those resources.

If the WAITERS increase as a result of a real deadlock situation, the deadlock must be found as quickly as possible. The larger the number of WAITERS, the longer deadlock processing takes to find and break the deadlock. While deadlock processing is running, **nothing else** is happening with respect to that particular IRLM! The effect on performance is particularly significant in a Sysplex environment because any IRLM that is running deadlock processing can cause other members to slow down or stop processing if they need this IRLM to process another request.

The previous paragraph might imply that you should have the DEADLOK parameter on the local IRLM set as high as possible, which would cause deadlock processing to run the least. However, if a deadlock situation exists, the faster it is detected and broken, the better the system will run.

**Recommendation:** Carefully chose a value for the deadlock frequency that is high enough so that deadlock processing runs often enough to resolve deadlock situations and at the same time is low enough so that system performance is minimally affected. If you have many threads that are capable of becoming WAITERS, it is extremely important to find and break that deadlock as quickly as possible before the WAIT chains require deadlock processing to run seconds (or even minutes) to perform a single local deadlock cycle. To find and break the deadlock is especially important in a Sysplex environment where multiple local cycles are required to break each deadlock, which in turn causes WAITERS to back up on all systems.

**Recommendation:** Monitor your system application programs for the actual number of deadlocks that IRLM had to break. If you find that the number of these deadlocks is low (or non-existent), you could raise the local deadlock value. If, however, you find that there are a lot of deadlocks that needed to be broken, you might consider having a low deadlock value.

Another aspect of deadlock processing and how it affects system performance is the number of locks that are held. IRLM must evaluate all held resources to check for WAITERS. Therefore, the amount of time that deadlock processing takes depends on the number of held resources (more held resources equals more deadlock processing time).

#### DESC=

Specifies the message descriptor code to be assigned to the IMS system console messages. If the parameter is not specified, then the value specified in the IMSCTF system definition macro is used.

The DESC parameter overrides the value specified on the IMSCTRL macro at system definition time.

**DIRCA=**

Specifies the size of the dependent region interregion communication area (DIRCA); the size specified must be a three-digit number (for example, 001) representing the number of 1-K blocks of subpool 251 to be reserved to hold a copy of your PCBs.

If you do not specify a DIRCA size, or if you specify a size of 000, IMS uses the system default for this message region. The size of the system default, determined during system initialization, is the maximum size required for any PSB. If necessary, IMS updates this default when an online change is made or when a dynamic PSB is scheduled.

Normally, you should not specify this parameter. You can use it to reduce the DIRCA size for a region where only PSBs with small DIRCA requirements are scheduled by class scheduling.

The output from message DFS6891 occurring during ACBGEN facilitates DIRCA size calculation. Each PSB has a PCB=value that is the sum of the PCBs within it. The PCB value plus 64 is the required DIRCA size, in bytes.

The Fast Path utility program uses a PSB that contains an I/O PCB and a DB PCB for the DBD specified in the JCL parameters. A DIRCA size of 002 is sufficient to contain the control blocks built for the PSB.

**DLIDSIZE**

Specifies how the full function processing uses its recovery data space. For a database recovery service, a data space is used to manage full-function updates. The specification indicates the amount of virtual storage for the full-function recovery data space that can be backed by real and expanded storage for those times when the full function recovery processing cannot keep up with the incoming log record stream.

If a number larger than the available storage for the Database Recovery Manager is specified, the machine can experience excessive paging, thus reducing the efficiency of the Database Recovery Manager. If a number smaller than the available storage for the database tracker is specified, recovery time may be affected as the speed that the Database Recovery Manager processes log data is slowed down.

**DSSIZE**

Specifies the maximum number of megabytes for DL/I database data set usage. The range of valid values is 15 to 1600. The default is 15 or the number of PSTs (set internally), whichever is greater.

**REDO** Specifies an estimated average size, in bytes, for the active subsystem's DL/I redo (type X'50') log records. The number specified will be rounded up to the nearest power of two. The range of valid values is 128 to 4096. The default is 256.

**DLINM=**

Specifies the member in SYS1.PROCLIB containing the procedure for the optional DL/I address space. If LSO=S is specified, DLINM= automatically starts the DL/I address space procedure. The name can be up to eight characters, the first of which must be alphabetic. The default is DLISAS.

**DLIPSB=**

When the DL/I address space option (LSO=S) is specified, two program specification block pools are used. CSAPSB specifies the size of pool in the

| z/OS common storage area (CSA), and DLIPSB specifies the size in the DL/I  
| address space. In an LSO=S system, the PSB= parameter is ignored.

The output of the ACBGEN utility indicates maximum CSA space, average CSA space, and the CSA space required by each PSB in the pool. This output is useful in determining what value to specify for the CSAPSB= parameter.

Similarly, before specifying the DLIPSB= parameter, you should consider the secondary address space (SAS) size required for each PSB, maximum SAS space, and average SAS space. Neither parameter should be zero. Normally, DLIPSB should be larger than CSAPSB.

The values for the pool sizes can be specified as either one- to six-numeric characters, or one- to five-numeric characters followed by either K (kilobyte), M (megabyte), or G (gigabyte). If K, M, or G is not specified, K is the default. The maximum value is 2G-1. The upper limit defaults to 2G-1 if any of the following statements is true:

- The value specified is not large enough to hold the largest primary and secondary storage allocations that are defined.
- The value specified exceeds 2G-1.
- No value is specified.

The sizes specified are rounded up to the nearest page boundary. The sizes of the two pools can also be specified during IMS system definition, using the SASPSB parameter in the BUFPOOLS macro.

For the FDR procedure, the sum of the values for the CSAPSB and DLIPSB defines the PSB pool size. If PSB is also specified, the larger value (PSB or the sum of CSAPSB and DLIPSB) is used.

#### **DLQT=**

Specifies a one- to three-digit number for the size of the dead letter queue, in days. Valid values are from 0 to 365. The default is 60. If you specify a null or invalid value, 60 is assumed.

#### **DMB=**

Specifies the amount of subpool 231 storage to be allocated to the DMB pool. The value can be specified as either one- to six-numeric characters, or one- to five-numeric characters followed by either K (kilobyte), M (megabyte), or G (gigabyte). If K, M, or G is not specified, K is the default. The maximum value is 2G-1. If the value specified exceeds 2G-1, the default is 2G-1. The pool size specified is rounded up to the nearest page boundary.

The output of the ACBGEN utility indicates the size of each DMB processed. This output should be examined before specifying the DMB= parameter.

#### **DMHVF=**

| Specifies (in megabytes) how much storage in the Fast Path DEDB VSO  
| emergency restart data space is to be page fixed. Valid values are from 0 to 99.

| This startup parameter is valid only on an XRF tracking system. The emergency  
| restart data space is deleted at the end of the XRF takeover. At that time, the  
| storage is also page freed.

#### **DPRTY=**

This parameter is ignored by OS/390, but retained in IMS for compatibility.

**Related Reading:** For details on DPRTY, see *MVS/ESA Job Control Language Reference*.

**DSCT=**

Specifies a one-character suffix identifying the user-supplied descriptor table, DFSDSCTy, in IMS.PROCLIB. The default is 0.

**EMHB= through FRE=****EMHB=**

Specifies a value for the upper expansion limit of the Fast Path expedited message handler buffer pool. The value can be specified as one- to six-numeric characters, or one- to five-numeric characters followed by K (kilobyte), M (megabyte), or G (gigabyte). If K, M, or G, is not specified, K is the default. The maximum value is 2G-1. The upper limit defaults to 2G-1 if any of the following statements is true:

- The value specified is not large enough to hold the largest primary and secondary storage allocations that are defined.
- The value specified exceeds 2G-1.
- No value is specified.

**EMHL=**

Specifies a one- to five-digit number for the length of the Fast Path expedited message handler buffer. If no buffer length is available from either the TRANSACT or APPL macro or, in the case of a static terminal, a FPBUF specification, this EMHL buffer length is used. Valid values are from 12 to 30720. The default is 2K. If you specify a null or invalid value, 2K is assumed.

**ENVIRON=**

Specifies the name of the PROCLIB member that contains the environment settings. A sample member DFSJVMEV will be supplied in the IMS sample library. ENVIRON is a required parameter.

The member named by the ENVIRON parameter can have comments but must begin with an asterisk (\*) in the first column. Each line in the options file must be no longer than 72 bytes including the continuation mark. To mark the continuation of a line, place the greater-than sign (>) at the end of the line. Path name strings are restricted to 255 bytes in length. A path name string can be one path name or several path names separated by a colon (:). The portion of the path name string greater than 255 bytes will be ignored.

**DEBUG= Y | N**

Specifies whether or not you want to run with the debug level of the JVM code. Specify DEBUG in the PROCLIB member that is named by the ENVIRON parameter.

**Y** IMS will load libjvm\_g.so (the debug level of the JVM code)

**N** N is the default value. IMS will not load the debug level of the JVM code.

**JLEOPT= Y | N**

Specifies whether (Y) or not (N) you want IMS to terminate the Language Environment (LE) enclave and reinitialize the LE enclave after each application program scheduling. This would enable (or not) the LE dynamic runtime parameter overrides for each application program scheduled in the JMP.

**LIBPATH=**

Required variable that must specify the environment variable. There must not be a space between the equal sign (=) in the LIBPATH= and the beginning of the first path name.



The LIBPATH= parameter must have a value equal to the following:

- The path name to your libjvm.so and libatoc.so DLLs.
- The path names to your IMS Version 9 Java class libraries (libJavTDLI.so and imsjava.jar). The path name to both of these Java class libraries should be /usr/lpp/ims/imsjava81 or equivalent.

For example, if you code ENVIRON=DFSENV on your JMP or JBP procedure, the DFSENV member must reside in IMS.PROCLIB or equivalent. The DFSENV member must contain the following (comments are optional):

```
*****
*LIBPATH environment variable*
*-----*
*/u/oeusr02/shiraz24/J1.3/bin/classic is path to libjvm.so*
*/u/oeusr02/shiraz24/J1.3/bin is path to libatoc.so*
*****
LIBPATH=/u/oeusr02/shiraz24/J1.3/bin/classic: >
/u/oeusr02/shiraz24/J1.3/bin:/usr/lpp/ims/imsjava81
```

### EPCB=

Specifies the amount of subpool 231 storage to be allocated to the EPCB pool. The value can be specified as one- to six-numeric characters or one- to five-numeric characters followed by K (kilobyte), M (megabyte), or G (gigabyte). If K, M, or G is not specified, K is the default. The maximum value that can be specified is 2G-1. If the value specified exceeds 2G-1 the default is 2G-1. The pool size specified is rounded up to the nearest page boundary. The size specified on the EPCB= parameter overrides the size specified in the BUFPOOLS macro.

If Fast Path is not generated, the EPCB pool is not allocated. If Fast Path is generated, an EPCB pool is required when the PSB, used by the IMS dependent region, has PCBs for Fast Path databases. This is true regardless of whether the region ever makes a call to these Fast Path databases. If nothing is specified on the EPCB= parameter or on the BUFPOOLS macro and if Fast Path is generated, the default pool size is 8K.

### ETO=

Specifies whether extended terminal option is enabled. The default is N.

- N** Specifies that terminals that are not defined to IMS have session initiation attempts rejected. Logon user data is not supported for static terminals; any user data provided at logon time is ignored.
- M** Specifies that terminals that are not defined to IMS have session establishment attempts rejected. Logon user data is supported for static terminals; any user data provided at logon time allows automatic signon to occur during session initiation.
- Y** Specifies that terminals that are not defined with system definition can establish sessions with IMS if the following conditions are true:
  - The terminal type is supported by IMS for dynamic allocation.
  - A default logon descriptor (provided either by IMS or by you) has sufficient information to create the control blocks necessary to accept the session request.
  - The ETO feature is installed.

### EXCPVR=

Specifies whether (1) or not (0) the OSAM Database Buffer Pool is to be page fixed. A value of 0 or 1 must appear in the generated JCL statement for this parameter.

**EXVR=**

Specifies whether (1 or Y) or not (0 or N) the Queue Manager buffer pools are to be long-term page fixed.

**FBP=**

Specifies the amount of subpool 0 storage to be allocated to the message format buffer pool. The value can be specified as one- to six-numeric characters or one- to five-numeric characters followed by K (kilobyte), M (megabyte), or G (gigabyte). If K, M, or G, is not specified, K is the default. The maximum value that can be specified is 2G-1. If the value specified exceeds 2G-1 the default is 2G-1. The pool size specified is rounded up to the nearest page boundary.

**FDRMBR=**

Specifies the two-character suffix for the FDR member DFSFDRxx in the IMS.PROCLIB.

**Restriction:** When you specify the FDR parameter on the IMS and DBCTL procedures you must also specify IRLM=Y.

**FESTIM=**

Specifies a timeout value in seconds to be used for front-end switching. The minimum value is 1, the maximum value is 300. If this parameter is not specified, the default is the value specified at system definition.

**FIX=**

Specifies the two-character suffix for DFSFIXxx and DFSDRFxx. This specifies the IMS.PROCLIB member to control both of the following:

- Page fixing portions of the control program
- Loading portions of the control program into DREF storage

For information on defining the content of these members, see “Defining Fixed Pages in the IMS Control Region” on page 352.

**FMTO=**

Specifies the type of dump output to be produced.

For a detailed explanation of this parameter, see Chapter 6, “Installing and Tailoring the Interactive Dump Formatter,” on page 315.

**FPDSSIZE**

Specifies the threshold value size of storage (in megabytes) used for the Fast Path recovery data space by ORS. When data space usage reaches the value specified, the stored update data is written to disk. Actual storage size can be larger than the specified value because additional data is normally processed while data is written to disk. The range of valid values is 1 to 1024. The default is 1 024.

**FPOPN=**

Specifies whether to reopen DEDB areas automatically during IMS restart.

**N** Specifies that areas should not be reopened automatically during IMS restart. Areas that are registered with DBRC for preopening will be preopened before the completion of an IMS start or restart as usual; other areas are not preopened.

This is the default.

**P** Specifies that after an IMS normal or emergency restart, IMS should preopen those areas that are registered with DBRC to be preopened asynchronously and concurrently with normal IMS online processing.

**R** Specifies that during IMS emergency restart, IMS should automatically reopen all areas that were open at the time of the previous IMS failure.

IMS reopens the areas after restart processing is complete. Reopening the areas occurs asynchronously and concurrently with normal IMS online processing.

**A** Specifies that during IMS normal or emergency restart, IMS should automatically reopen all areas that were open at the time of the previous IMS failure. This option also specifies that IMS should preopen those areas that are registered with DBRC to be preopened.

IMS reopens the areas after completing restart processing. Reopening and preopening the areas occurs asynchronously and concurrently with normal IMS online processing.

**D** Specifies that during IMS control region initialization, the DEDB area preopen process is disabled. Areas that are defined to DBRC with the PREOPEN option are not opened. The open process occurs when the dependent region first requests data or when a /STA AREA command is issued after restart is complete. This option is valid for an IMS cold start, normal restart, and emergency restart. This option has no effect on an XRF tracking system or an FDBR system.

The FPOPN option does not change any area options in DBRC. This includes PREOPEN, VSO, and PRELOAD. These options remain set in the RECON data set.

This option does not affect DEDB forward recovery (REDO) processing. If an area requires forward recovery during an emergency restart, the area is opened, updated, and then closed.

#### **FPRLM**

Specifies whether to restart DEDB areas automatically during IRLM reconnect.

**N** Specifies that areas should not be restarted automatically during IRLM reconnect.

This is the default.

**P** Specifies that after IRLM reconnect, IMS should preopen those areas that are registered with DBRC to be preopened asynchronously and concurrently with normal IMS online processing.

**S** Specifies that after IRLM reconnect, IMS should automatically restart all areas that were started at the time of the IRLM failure.

IMS restarts the areas after IRLM completes reconnect processing. Restarting the areas occurs asynchronously and concurrently with normal IMS online processing.

**R** Specifies that during IRLM reconnect, IMS should automatically restart and reopen all areas that were open and started at the time of the IRLM failure.

IMS restarts the areas after IRLM completes reconnect processing. Restarting the areas occurs asynchronously and concurrently with normal IMS online processing.

After IMS restarts all of the areas that were started at the time of the IRLM failure, IMS reopens them. If an area is not yet reopened when an IMS application program requests access, it will be reopened immediately. Restarting and reopening the areas occurs asynchronously and concurrently with normal IMS online processing.

**FPWP=**

Specifies a value representing the upper expansion limit of the Fast Path work pool. The value can be specified as one- to six-numeric characters or one- to five-numeric characters followed by K (kilobyte), M (megabyte), or G (gigabyte). If K, M, or G is not specified, K is the default. The maximum value that can be specified is 2G-1. If the value specified exceeds 2G-1 the default is 2G-1. If a value is not specified, the upper limit default is 2G-1.

**FRE=**

Specifies a one- to five-digit number of fetch request elements used to load MFS blocks into the message format buffer pool.

**GRNAME= through ISIS=****GRNAME=**

Specifies a one- to eight-character name of the XCF group for IMS Open Transaction Manager Access (OTMA). The group name should be alphanumeric (upper case only) or specified with the following special characters: #, \$, @.

The member name IMS uses when joining the XCF group comes from the USERVAR specification or defaults to the IMS APPLID specification.

This parameter is only valid in IMS DB/DC or IMS TM-DB2 environments.

**GRSNAME**

Specifies a one- to eight-byte character name of the generic resources group. The group name should be specified in upper case alphanumeric characters, or with the following special characters: #, \$, @. GRSNAME must begin with an alphabetic character. You must specify the same GRSNAME for all IMS systems participating within the same generic resources group.

**GSGNAME=**

Specifies the global service group name to be used. If GSGNAME= is not specified in the DBBATCH procedure, the default is the GSGNAME specified in the IMSCTRL macro.

If GSGNAME=NONE is specified, this procedure does not reference any global service group and any activity performed is not tracked by RSR.

If a GSGNAME is supplied either by the GSGNAME= specification or from the system definition IMSCTRL macro, then *DBRC=Y* must be used.

**HASH=**

Specifies the number of hash entries available in the lock hash table of the coupling facility lock structure (in units of 1048576). The lock hash table is used by IRLM as an index into the lock table. HASH can have a value of blank, zero, or any exact power of 2, up to a maximum of 1024. For example, a value of HASH=32 would result in a hash table size of 64 MB, assuming the width of each hash entry to be 2 bytes. If the value specified on HASH is incorrect, the START will terminate with DXR116E CODE=24 and ABENDDU2018. This value is only used if SCOPE=GLOBAL or NODISCON has a default of blank (IRLM calculates the value). The number of hash entries for the group is used in the following order and the width is controlled by the value specified on MAXUSERS. Both of these are dictated by the first IRLM to connect to the group during initial structure allocation or during a REBUILD:

1. The HASH value specified in the MODIFY ir1mproc,SET,HASH= command, if greater than zero.
2. The value from the HASH= parameter in the IRMLPROC, if greater than zero.

- The existing logic, which determines the nearest power of 2, after dividing the XES structure size (returned on the IXCQUERY call) by 2 times the HASH width (based on the value of MAXUSERS).

If an attempt is made to use a nonzero value from either number 1 or number 2 in the previous list and the value would require more storage than is available in the structure size (returned by the XES IXCQUERY request issued by IRLM), then the value from the next priority order is used (number 2 or number 3 from the previous list). IRLM does not try to determine how many record table entries the user wants. If the HASH= value consumes most or all of the coupling facility lock structure size available, IRLM will allow the connect to succeed.

Table 29 illustrates some of the commonly desired values for lock table hash entries and the resulting storage that would be needed for the lock hash table. The storage available for record table entries is the INITSIZE minus the storage listed in the following table.

Table 29. Hash Value and Resulting Lock Table Storage

For HASH=	Lock Table Storage if 2-Byte Entries	Lock Table Storage if 4-Byte Entries
8	16MB	32MB
16	32MB	64MB
32	64MB	128MB
64	128MB	256MB
128	256MB	512MB
256	521MB	1024MB

#### HIOP=

Specifies a value for the upper expansion limit of the high communication I/O pool. The value can be specified as one- to six-numeric characters or one- to five-numeric characters followed by K (kilobyte), M (megabyte), or G (gigabyte). If K, M, or G is not specified, K is the default. The maximum value that can be specified is 2G-1. If the value specified exceeds 2G-1, the default is 2G-1. If a value is not specified, the upper limit default is 2G-1.

#### HSBID=

Specifies (for XRF systems only) the XRF system identification. One system is associated with digit 1 and one is associated with digit 2. Either can be brought up as an active or an alternate system.

Through this parameter, IMS identifies the master terminal and the message queues that are associated with it. For example, an IMS system with HSBID=1 uses the first master terminal in the stage 1 definition.

It is recommended that you code the HSBID parameter if you are planning to run XRF anytime in the future. Having HSBID coded could save performing an IMS definition at a later date.

XRF requires that two APPLIDs be specified on the system definition COMM macro. XRF also requires that two *nodenames* be specified by the NAME parameter on the TERMINAL macro. These are for the master and secondary master terminal.

If a null value is specified for HSBID (for example, HSBID= ' '), XRF is disabled. Changing the HSBID from usable ('1' or '2') to null or from null to usable requires a cold start.

**HSBMBR=**

Specifies (for XRF systems only) the two-character suffix for XRF member DFSHSBxx in the IMS.PROCLIB. The default is 00.

**IMSGROUP=**

Specifies a one- to four-character IMS group name. This parameter is user-generated and is usually the IMSID value most often found in the BMP or other dependent region PROCLIB members. There is no default value. If no IMS group name is specified, the z/OS name token is *not* built.

The IMSGROUP= parameter must equal the same value as the IMSID of the BMP regions.

The IMSID of the CTL region should be different because it must be unique within the sysplex. Using the IMSID of the BMP regions as the IMSGROUP= value in the CTL region is to eliminate the need to add any new parameters to the BMP (and other dependent region's) JCL.

Example:

```
CTL Region 1 IMSID=IMS1, IMSGROUP=IMSB
CTL Region 2 IMSID=IMS2, IMSGROUP=IMSB
BMP Region 1 IMSID=IMSB
MPP Region 1 IMSID=IMSB
```

In the example, BMP region 1 connects to either IMS1 or 2 using z/OS names services to find the correct IMSID to use. The IMSID of the BMP JCL is used to construct the z/OS name. This example also holds true for the MPP region and any IFP regions.

The one EXEC parameter, IMSGROUP= is added to the control region procedure.

**IMSID=**

**For IMS control region:** Specifies a one- to four-character identifier that is a valid subsystem identifier to the operating system being used. It overrides the identifier specified at the time of system definition of the running IMS system.

**For IMS dependent regions:** Specifies a one- to four-character identifier that is a valid subsystem identifier to the operating system to which this dependent region will connect. It overrides the identifier specified at the time of system definition of the running IMS system.

**For IMS batch region:** Specifies a one- to four-character IMS identifier that is used in IMS messages that are written to the system log. It overrides the identifier specified at the time of system definition of the running IMS system.

Do not use characters for the IMSID that match the beginning characters of an z/OS command. If the IMSID is the same as the beginning characters of an z/OS command, that command does not work after you start IMS. For example, if you start IMS with IMSID=D, z/OS does not respond to any of the z/OS display commands such as D A,L.

The IMSID name must not be the same as the procedure name that starts IMS unless one of the following is true:

- All DD statements in the startup procedure are cataloged in the master catalog.
- The unit and volume are specified on each DD statement.

This parameter cannot be changed at emergency restart.

An IMSID name specified in the FDR procedure identifies the subsystem name for the FDR region.

**IMSPLEX=cccc**

Specifies the five-character IMSplex group name, left-justified and padded with blanks, if necessary. The IMSplex group name is passed to the SCI registration exit routine DSPSCIX0. The sample version of DSPSCIX0 shipped with IMS will return the value that you supply to DBRC as the IMSplex name.

**IN=**

Specifies an input transaction code. This parameter is necessary only when the application program intends to access the message queues. If this parameter is specified, the OUT= parameter is ignored. You cannot schedule a BMP with an IN= parameter specified against a PDIR with an associated SMB that is already scheduled and is SCHDTYP=SERIAL. Doing so results in ABENDU0457 when the BMP is scheduled.

**IOB=**

This parameter is no longer used. Requests for I/O are now dynamically allocated.

**IOVFI=**

Specifies how often the 'count of unused IOVF control intervals' is updated. This parameter sets a timer, in seconds, which triggers an IMS internal task to begin a count of unused IOVF control intervals.

The default value is 7200 seconds (2 hours). To disable the timer, specify a time of 1 second. A value of 0 sets the timer to the default value (7200 seconds). The maximum allowed value is 86400 (24 hours).

If the counter task is currently executing and the time interval expires to initiate another counter task, the second count request will be ignored. The start time, stop time, and accumulate number of runs and corresponding total time for these runs are tracked in the 59FF logrec.

**IRLM=**

Specifies whether you want to use the IRLM during this execution. The default for IRLM= depends on the values you specify in the IRLM= and the IRLMNM= parameters of the IMSCTRL macro. If IRLMNM= is not specified on the IMSCTRL macro or on the execution JCL and IRLM= is specified on the execution JCL, then the IRLM name used is *IRLM*.

This parameter cannot be changed at emergency restart.

**Y** Indicates that you will use IRLM. This is the default under either of the following conditions:

- The IMSCTRL macro specifies IRLM=Y.
- The IMSCTRL macro specifies an IRLMNM and IRLM=N is **not** specified.

**N** Indicates that you will not use IRLM. This is the default under either of the following conditions:

- The IMSCTRL macro specifies IRLM=N.
- The IRLM= and IRLMNM= keywords of the IMSCTRL macro do not specify a value.

**LTE**

Specifies the number of lock table entries available in the coupling facility lock structure (in units of 1048576). LTE can have a value of blank, zero, or any exact power of 2 up to a maximum of 1024.

For example, a value of `LTE=32` would result in a lock table size of 64MB, assuming a width of 2 bytes for each lock table entry. If the value specified on `LTE` is incorrect, `START` will terminate with `DXR116E CODE=24` and `ABENDU2018`. This value is only used if `SCOPE=GLOBAL` or `NODISCON` and has a default of blank (IRLM calculates the value). The number of lock table entries in the group is used in the following order, and the width is controlled by the value specified on `MAXUSRS`. Both of these are dictated by the first IRLM to connect to the group during initial structure allocation or during a `REBUILD`:

- 1 The value specified on the `MODIFY ir1mproc,SET,LTE=` command if greater than zero.
- 2 The value from the `LTE=` in the `IRLMPROC` if greater than zero.
- 3 The existing logic, which determines the nearest power of 2 after dividing the XES structure size returned on the `IXCQUERY` call by 2 times the `LTE` width (based on `MAXUSRS`).

If an attempt is made to use a nonzero value from either number 1 or number 2 and that value would require more storage than is available in the structure size returned by the XES `IXCQUERY` issued by IRLM, then the value from the next priority order is used (number 2 or number 3 from the previous list). IRLM does not try to determine how many record table entries the user wants. If the `LTE=` value consumes most or all of the coupling facility lock structure size available, IRLM will allow the connect to succeed.

Table 30 illustrates some of the commonly desired values for lock table entries and the `STORAGE` that would be needed for the `LOCK TABLE` portion as a result. The storage available for record table entries is the `INITSIZE` minus the storage listed in the following table.

Table 30. Common Lock Table Entry values

For <code>LTE=</code>	Storage if 2-byte entries	Storage if 4-byte entries
8	16MB	32MB
16	32MB	64MB
32	64MB	128MB
64	128MB	256MB
128	256MB	512MB
256	512MB	1024MB

#### **IRLMGRP=**

Specifies the name of the XCF group to which this IRLM belongs. All IRLMs in the same group must specify the same `LOCKTABL` parameter and an `IRLMID` value that is unique within the group. The default is `IRLMDS`.

The group name (`IRLMDS` by default) is used as the XCF group name. This name cannot start with "SYS" and cannot be the same as the `LOCKTABL` parameter.

#### **IRLMID=**

Specifies a decimal number that is used to distinguish between IRLMs in a data sharing group. The IRLM with the lowest ID number in the group becomes the



global deadlock manager for the group when you are in a data sharing module. There is no default value. A unique value must be specified for each IRLM in the data sharing group.

You can specify this parameter as either a one- to three-digit number from 1 to 255, or as a printable character. You must enclose a printable character with seven single quotation marks on either side of the character. Thus, you must specify the character *D* as `IRLMID='''D'''`. When you specify a printable character, IRLM uses the EBCDIC value of the character as the IRLMID.

**IRLMNM=**

Specifies the four-byte z/OS subsystem name assigned to this IRLM. IRLM requires a four-byte name for its internal processing, even though z/OS can accept names with fewer than four bytes for subsystem names.

**ISIS=**

Specifies whether application group name security or resource access security checking is to be performed. If this parameter is not specified, the default specification on the TYPE parameter of the SECURITY macro is used.

- |  |          |   |
|--|----------|---|
|  | <b>0</b> | Specifies that no application group name security is to be performed.   |
|  | <b>1</b> | Specifies that application group name security checking using RACF is to be performed.                        |
|  | <b>2</b> | Specifies that application group name security checking using a user-written exit routine is to be performed. |
|  | <b>N</b> | Specifies that no application group name security or resource access security is to be performed.             |
|  | <b>R</b> | Specifies that resource access security checking using RACF is to be performed.                               |
|  | <b>C</b> | Specifies that resource access security checking using a user exit routine is to be performed.                |
|  | <b>A</b> | Specifies that resource access security checking using both RACF and a user exit routine is to be performed.  |

This parameter cannot be changed at emergency restart.

**JVMOPMAS= through LUMP=****JVMOPMAS=**

Sets the name of the member in IMS.PROCLIB that contains the JVM options for the standalone JVM for JBP regions. *member name* is a maximum of 8 uppercase characters.

This JVM options member must contain the following:

- `-Dibm.jvm.shareable.application.class.path=`  
Specify the path name (or path names) of your IMS Java application class files. If your .class files are contained in a .jar file, the path name to the .jar file must be fully qualified, including the name of the .jar file.
- `-Dibm.jvm.trusted.middleware.class.path=`  
Specify at least the path name of your `imsjava.jar` file.

Comments are supported for this options member. The comments begin with an asterisk (\*) in the first column.

Each line in the options file must not be longer than 72 bytes, including the continuation mark. Use a greater-than symbol (>) at the end of the line as a continuation character.

Path strings can be a maximum of 255 bytes in length (any characters over 255 bytes will be ignored). A path string can be one path name or several path names. If you are specifying multiple path names, separate each by a colon (:).

The sample member, DFSJVMMS, demonstrates how to specify the JVM options for JBP regions.

#### **JVMOPWKR=**

Sets the name of the member in IMS.PROCLIB that contains the JVM options for the worker JVM for JMP regions. JVMOPWKR is an optional parameter.

Comments are supported for this options member. The comments begin with an asterisk (\*) in the first column.

Each line in the options file must not be longer than 72 bytes, including the continuation mark. Use a greater-than symbol (>) at the end of the line as a continuation character.

Path strings can be a maximum of 255 bytes in length (any characters over 255 bytes will be ignored). A path string can be one path name or several path names. If you are specifying multiple path names, separate each by a colon (:).

The sample member, DFSJVMWK, demonstrates how to specify the JVM options for JBP regions.

#### **LGMSGSZ=**

Specifies a one- to five-digit number that represents the size in bytes of a long message record. The length specified is rounded up to the nearest multiple of four.

**Restriction:** The length must be greater than or equal to the length specified for the short message record or one and one-half times the maximum message prefix size plus four. If you specify a value less than the short message record length or one and one-half times the maximum prefix size plus four, the length of the large message size is set to the greater value between the large message DCB LRECL length, the short message size, or one and one-half times the maximum message prefix size plus four. The length of the long message record cannot exceed 30632 bytes. If you specify a value greater than 30632, the size is set to 30632.

This parameter is valid only in the Shared Queues environment. If you do not specify a value for the SHAREDQ= parameter, the value specified for LGMSGSZ= is ignored. If you specify a value for the SHAREDQ= parameter, but not for the LGMSGSZ= parameter, the size of the long message record is obtained from whichever is greater: the DCB generated for the long message queue data set or the short message record length.

#### **LGNR=**

Specifies the maximum number of Fast Path DEDB buffer alterations that is to be held before the entire control interval is logged. The parameter is used by Fast Path to build the Fast Path buffer header (DMHR) and to control the DEDB update logging mechanism. The number of DEDB buffer alterations has no direct relationship to the number of DEDB calls. The parameter can be specified as a one- to two-digit numeric value. If a value less than seven is specified, a minimum value of seven is set.

**LHTS=**

Specifies a one- to five-digit number of LTERM hash table slots. Valid values are from 0 to 32767. The default is 256. If you specify a null or invalid value, 256 is assumed. For more information see "Specifying the Number of Hash Table Slots" on page 462.

**LOCKMAX=**

Specifies a value between 1- and 32767- (in units of 1000). This parameter overrides the PSBGEN LOCKMAX value if one was specified. An override parameter of LOCKMAX=0 turns off all locking limitation.

**LOCKMAX=10**

Allows for 10000 locks.

**LOCKMAX=0**

Turns off locking limitation.

**LOCKTAB=**

This parameter is optional and specifies the lock table to be used by this group. The lock table must have previously been defined through XCF Resource Manager panels and must be an element of the currently active locking policy. The default is IRLMLT1.

This parameter must be the same for all IRLMs that specify the same value for the GROUP parameter.

**LOGA=**

This parameter is no longer used and is ignored. In earlier releases, it specified whether IMS was to use the BSAM (0) or OSAM (1) logging access method.

**LOGT=**

Specifies the tape device type where the log data set is to be mounted. The tape device type specified by the LOGT= parameter substitutes for the device parameter specified in the IEFORDER DD statement. The default is device type 2400.

**LSO=**

Specifies whether the Local Storage Option is to be used. When it is used, some IMS modules and buffers are moved from the CSA to the private storage area of the control region.

**Y** Specifies that LSO is to be used. This is the default.

**S** Specifies the DL/I subordinate address space option. This is required for a CCTL connected to an IMS control region or DBCTL region.

If LSO=Y is specified for an RSR tracking subsystem, DL/I database tracking is not initialized; only Fast Path database tracking can be performed.

**LTE**

Specifies the number of lock table entries available in the coupling facility lock structure (in units of 1048576). LTE can have a value of blank, zero, or any exact power of 2 up to a maximum of 1024.

For example, a value of LTE=32 would result in a lock table size of 64MB, assuming a width of 2 bytes for each lock table entry. If the value specified on LTE is incorrect, START will terminate with DXR116E CODE=24 and ABENDU2018. This value is only used if SCOPE=GLOBAL or NODISCON and has a default of blank (IRLM calculates the value). The number of lock table entries in the group is used in the following order, and the width is controlled by the value specified on MAXUSRS. Both of these are dictated by the first IRLM to connect to the group during initial structure allocation or during a REBUILD:

- 1 The value specified on the MODIFY ir1mproc,SET,LTE= command if greater than zero.
- 2 The value from the LTE= in the IRLMPROC if greater than zero.
- 3 The existing logic, which determines the nearest power of 2 after dividing the XES structure size returned on the IXCQUERY call by 2 times the LTE width (based on MAXUSRS).

If an attempt is made to use a nonzero value from either number 1 or number 2 and that value would require more storage than is available in the structure size returned by the XES IXCQUERY issued by IRLM, then the value from the next priority order is used (number 2 or number 3 from the previous list). IRLM does not try to determine how many record table entries the user wants. If the LTE= value consumes most or all of the coupling facility lock structure size available, IRLM will allow the connect to succeed.

Table 30 on page 278 illustrates some of the commonly desired values for lock table entries and the STORAGE that would be needed for the LOCK TABLE portion as a result. The storage available for record table entries is the INITSIZE minus the storage listed in the following table.

Table 31. Common Lock Table Entry values

For LTE=	Storage if 2-byte entries	Storage if 4-byte entries
8	16MB	32MB
16	32MB	64MB
32	64MB	128MB
64	128MB	256MB
128	256MB	512MB
256	512MB	1024MB

#### LTERM=

Specifies whether the LTERM name of the static input terminal is used in the DFSAPPC. Y the default, specifies the LTERM is used. N specifies the LTERM is not used. If LTERM=N is specified, and a user ID is not provided or a user is not signed on, the DFSAPPC process is rejected with an error message DFS1957E.

#### LUMC=

Specifies a value for the upper expansion limit of the LU 6.2 device manager common buffer pool. The value can be specified as one- to six-numeric characters or one- to five-numeric characters followed by K (kilobyte), M (megabyte), or G (gigabyte). If K, M, or G is not specified, K is the default. The maximum value is 2G-1. The upper limit defaults to 2G-1 if any of the following statements is true:

- The value specified is not large enough to hold the largest primary and secondary storage allocations that are defined.
- The value specified exceeds 2G-1.
- No value is specified.

The OCO LUM code uses the pool for work area processing.

#### LUMP=

Specifies value for the upper expansion limit of the LU 6.2 device manager private buffer pool. The value can be specified as one- to six-numeric

characters, or one- to five-numeric characters followed by K (kilobyte), M (megabyte), or G (gigabyte). If K, M, or G is not specified, K is the default. The maximum value is 2G-1. The upper limit defaults to 2G-1 if any of the following statements is true:

- The value specified is not large enough to hold the largest primary and secondary storage allocations that are defined.
- The value specified exceeds 2G-1.
- No value is specified.

The OCO LUM code uses the pool for work area processing when not processing in the dependent region.

## MAXCSA= through OVLA=

### MAXCSA=

The DXRJPROC sample member has a DEFAULT of 6M. The acceptable value range is 1M - 999M. Specify the maximum amount of common service area (CSA) and extended CSA (ECSA) that the IRLM for this IMS uses for its lock control block structures. 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 (such as 5000K) and M for megabytes (such as 5M).

**Note:** This parameter may be modified with the following command: `MODIFY ir1mproc,SET,CSA=nnn.`

### MAXPST=

Specifies the maximum number of blocks that the user permits the online control region to allocate. The default is 255; the maximum is 999. When system activity increases, IMS allocates PST blocks up to the maximum specified. When the workload diminishes, these blocks are unallocated and storage is released to as low as the value specified by PST=. An example of how these parameters can be specified follows:

#### Example:

```
...,PARM='PST=48,MAXPST=300,...
```

**Restriction:** Reducing the value of the MAXPST parameter from a previous IMS start requires a coldstart. Otherwise, results are unpredictable.

### MAXUSRS=

Specifies the maximum number of IRLMs that are to connect to the data sharing group. A value from 2 to 248 can be specified.

The initial allocation of locking resources in XES is dependent on the number of entries and the maximum number of users, so some assessment of available resources is made when IRLM joins the group. However, nothing is done to keep the number of users below the specified value.

### MBR=

Specifies an application program name.

### MCS=(x,y)

Specifies the z/OS routing code to be assigned to the IMS system console if multiple console support (MCS) is included in the operating system. If the parameter is not specified, then the value specified in the IMSCTF macro is used.

The MCS parameter overrides the value specified on the IMSCTRL macro at system definition time.

Although z/OS supports more than 16 route codes, IMS uses only route codes 1 through 16.

**MNPS=*name***

Specifies the name of the MNPS ACB to use for terminal recovery. This parameter is valid only for XRF systems. If specified:

- This parameter overrides the MNPS name specified on the DFSHSBxx PROCLIB member.
- The USERVAR specification is ignored.

This parameter is optional.

**MNPSPW=*name***

Specifies the password to use for the MNPS ACB. If specified, this parameter overrides the MNPSPW name specified on the DFSHSBxx IMS.PROCLIB member. VTAM checks this password. If VTAM requires a password during VTAM system definition and no password is specified, then the MNPS ACB cannot be opened.

**MON=**

Specifies whether (Y) or not (N) the IMS monitor is to be active for this execution.

**MSDB=**

Specifies the one-character suffix for DBFMSDBx. (See “Member DBFMSDBx” on page 333.)

**NBA=**

A four-digit number specifying the number of Fast Path database buffers to be made available in the Common Service Area when the Fast Path region is activated. These buffers are used by the DEDBs to access VSAM control intervals, and by the MSDBs and DEDBs to hold information for updating between the DL/I call and synchronization point.

These buffers become part of the system database resources, but the number specified is reserved for exclusive use by this region when the application program accesses the MSDBs or DEDBs. (Although you are guaranteed access to a number of buffers, a maximum of the NBA + OBA value, you might have to wait for them.) The default is 0.

**NHTS=**

Specifies a one- to five-digit number of terminal hash table slots. Valid values are from 0 to 32767. The default is 256. If you specify a null or invalid value, 256 is assumed. For more information see “Specifying the Number of Hash Table Slots” on page 462.

**NLXB=**

Specifies the number of parallel sessions to be added during the startup of an IMS system. The NLXB value is added to the value specified in the SESSION parameter during system definition to increase the number of link extension blocks (LXB) generated for each link control block (LCB). The default for NLXB is 0. If NLXB is specified when the SESSION value is already set at the maximum (676), the NLXB value is ignored. This parameter is valid for MSC with VTAM only and does not apply to Intersystem Communication (ISC).

**OBA=**

A four-digit number specifying the number of additional page-fixed buffers to be made available to a Fast Path application region if the normal allotment is used.

In the Common Service Area, the system page fixes only enough additional buffers to handle the largest overflow specification from all active regions. The default is 0.

The system only allows one program at a time to use its overflow allocation. This is done to prevent a situation in which every program is waiting for buffers, when none is available. If an application needs more buffers than the sum of NBA + OBA, it abnormally terminates (for message-driven programs) or an FR status code is issued (for nonmessage-driven programs).

**ODBASE=**

Specifies whether (Y) or not (N) SAF security checking is to be performed on an APSB request form an ODBA thread. The default is N.

**Y** specifies that a SAF RACROUTE AUTH call is to be made. The AIMS class is used to verify that the user can access the PSB.

**N** specifies that so SAF calls are to be made.

**OPT=**

Specifies what action to take when the dependent region starts, but when no system identifier matches the names given on IMSID or ALTID. The actions are:

**W** Wait for the control program to start.

**N** Ask operator for a decision. This is the default.

**C** Cancel message region automatically.

**ORS=**

Identifies this address space as part of ORS to DFSMVRC0 so that the Recovery Data Manager is started.

**ORSMBR=**

Specifies the two-character suffix for the ORS member (DFSORSxx) in IMS.PROCLIB that is currently in effect. There is no default.

**OTHR=**

Specifies the number of concurrent output threads that Fast Path is to support for the entire Fast Path system. This can be any value from 1 to 255, but no more than the value specified on the MAXPST= parameter of the IMS procedure. OTHR= *n* causes a total of *n* service request blocks (SRBs) and extended service request blocks (ESRBs) to be created at system initialization. These blocks are used in scheduling asynchronous DEDB output. If an insufficient number of SRBs is specified, write buffers are queued until one becomes available.

**OTMA=**

Specifies that the IMS Open Transaction Manager Access (OTMA) function is to be enabled during IMS initialization. Valid values are Y (yes) or N (no). The default value is N.

If Y is specified, IMS attempts to create an XCF group, specified by GRNAME, during initialization and then attempts to join that group.

When OTMA=Y is specified, IMS uses the security level that is currently in effect (CHECK, FULL, NONE, or PROFILE) to control the RACF security level for input from IMS OTMA clients. You can issue the /DISPLAY OTMA command to show the security level that is currently in effect.

After an IMS cold start, the security default is FULL. If the IMS initialization parameter OTMASE is not specified, IMS retains OTMA security settings (which are established by the /SECURE OTMA command) after a warm start or

emergency restart. If the OTMASE parameter is specified, the security option for OTMA is determined by the option that is specified on the OTMASE= parameter.

The OTMA parameter is only valid in IMS DB/DC or DCCTL environments.

#### **OTMAASY= Y | N**

Specifies that a non-response transaction originating from a program-to-program switch should be scheduled asynchronously. Valid values are Y(yes) or N (no). The default value is N.

This parameter is for Send-then-commit messages only. There is no DFS2082 issued for a transaction scheduled asynchronously. This parameter can also be used in the multiple program-to-program switches environment to ensure that only response transactions can be scheduled synchronously

#### **OTMAMD=**

Specifies whether the member override function of the OTMA Prerouting Exit routine (DFSYPX0) is enabled for a transaction initiated from an OTMA client. Valid values are Y(yes) or N(no). The default value is N.

#### **OTMANM=**

Specifies the member name IMS uses when joining the XCF group for non-RSR or non-XRF systems.

#### **OTMASE=**

Specifies the type of OTMA RACF security that you want to use, if any. Valid parameter values include:

- C** OTMA RACF security is CHECK. IMS commands are checked against the CIMS class. IMS transactions are checked against the TIMS class.
- F** OTMA RACF security is FULL. This provides the same type of security as CHECK, but additional security checking is performed against dependent regions. X is the default value for the OTMASE parameter if no value is specified. X provides OTMA RACF security that is similar to F (FULL).
- N** OTMA RACF security is NONE. No calls to RACF are made.
- P** OTMA RACF security is PROFILE. Each OTMA message defines the level of security checking to be done.

The /SECURE OTMA command overrides the value you specify in the OTMASE= keyword.

The default value is F.

If you do not specify the OTMASE keyword, IMS retains the OTMA security settings (which are established by the /SECURE OTMA command) after a warm start or an emergency restart.

#### **OTMASP=**

Specifies whether a non-synchronous Tpipe or a synchronous Tpipe will be created to deliver the OTMA output. Valid values are Y (create a synchronous Tpipe) or N(create a non-synchronous Tpipe). The default value is N. If the value of this parameter is Y or if the OTMA Prerouting Exit routine (DFSYDRU0) indicates the need for a synchronous Tpipe, the synchronous Tpipe will be created.

#### **OUT=**

Specifies the transaction code or logical terminal name to which an output



message is to be sent. This parameter is necessary when the application program wants to send output without accessing the input queues. A remote LTERM (CNT) cannot be specified.

This parameter is ignored if IN= is also specified.

#### **OVLA=**

Specifies the overlay supervisor option:

- 0** Allow z/OS to load and delete the overlay supervisor for every overlay application program. This is the default.
- 1** Load and retain a copy of the overlay supervisor when the message region is initialized.

## **PAGES= through PWFI=**

#### **PAGES=**

Specifies the amount of primary and secondary space allocated for the SYSPRINT DD statement data set.

#### **PARDLI=**

Specifies the parallel DL/I option.

- 0** DL/I processing is to be performed within the BMP region. This is the default.
- 1** All DL/I processing for this BMP is to be performed in the IMS control region.

PARDLI=1 prevents control region system 113 abends resulting from system X22 abends in the BMP region. If PARDLI=1, parallel DL/I is disabled. This can degrade performance.

#### **PASSWD=**

This parameter is obsolete. Use the PASSWD1 parameter.

#### **PASSWD1=**

Specifies a password to override the value specified on the PASSWD= parameter in the COMM macro.

#### **PC=**

If PC=YES is specified, the IRLM lock control block structures reside in the IRLM address space, and the space switching program call (PC) instruction is used to obtain addressability to the structures. The MAXCSA value is ignored.

If PC= NO is specified, the IRLM lock structures reside in ECSA. The default for this parameter is PC=NO. The MAXCSA value is used to limit the amount of CSA + ECSA used by IRLM for lock structures, and is considered 'accountable' storage in the STATUS display. However, IRLM can use CSA & ECSA beyond this value for other required structures. This is considered 'non-accountable' in the STATUS display.

If the value NO is specified, the IRLM lock structure resides in CSA. The default for this parameter is PC=NO.

#### **PCB=**

Specifies the size of the dependent region interregion communication area (DIRCA); the size specified must be a three-digit number (for example, 001) representing the number of 1 KB blocks of subpool 251 to be reserved to hold a copy of your PCBs.

If you do not specify a DIRCA size or if you specify a size of 000, IMS uses the system default for this message region. The size of the system default,

determined during system initialization, is the maximum size required for any PSB. If necessary, this system default is updated when an online change is made or when a dynamic PSB is scheduled.

If you specify this parameter, the specified size is always used. If the size required is larger than the specified size, the region abends with a user 0242 abend code.

Normally, you should not specify this parameter. You can use it to reduce the DIRCA size for a region when PSBs with small DIRCA requirements are scheduled into this region.

The output from message DFS589I occurring during ACBGEN facilitates DIRCA size calculation. Each PSB has a PCB= value that is the sum of the PCBs within it. The PCB value + 64 becomes the required DIRCA in bytes.

**PGPROT=**

Specifies whether the IRLM is to place load modules that are resident in common storage into z/OS Page Protected Storage during initialization. If the load modules are in Page Protected Storage, any application that attempts to overlay the modules is terminated. The default value is YES.

**PIINCR=**

Specifies the increment to be used in a conditional GETMAIN to obtain the dynamic storage. The default value is 64K.

No storage is allocated for ENQUEUE/DEQUEUE blocks at initialization. The first block of storage, obtained when the ENQUEUE/DEQUEUE routine is first called, is in the increment specified in PIINCR. Storage continues to be obtained in the increment specified until the maximum is reached (see PIMAX=). IMS does not release any storage that is dynamically obtained for the ENQUEUE/DEQUEUE routine. After storage is obtained, it is kept for the duration of the control region execution.

PIINCR can be specified as either one- to six-numeric characters, or one- to five-numeric characters followed by K (kilobyte), M (megabyte), or G (gigabyte). If K, M, or G is not specified, K is the default. The maximum value is 2G-1. The upper limit defaults to 2G-1 if any of the following statements is true:

- The value specified is not large enough to hold the largest primary and secondary storage allocations that are defined.
- The value specified exceeds 2G-1.
- No value is specified.

**PIMAX=**

Specifies the maximum amount of dynamic storage available to the exclusive control of the ENQUEUE/DEQUEUE routine. The default value is 1024K (1M).

No storage is allocated for ENQUEUE/DEQUEUE blocks at initialization. The first block of storage, obtained when the ENQUEUE/DEQUEUE routine is first called, is in the increment specified by PIINCR=. Storage continues to be obtained in the increment specified until the maximum is reached. IMS does not release any storage that is dynamically obtained for the ENQUEUE/DEQUEUE routine. After storage is obtained, it is kept for the duration of the control region execution.

PIMAX can be specified as either one- to six-numeric characters, or one- to five-numeric characters followed by either K (kilobyte), M (megabyte), or G (gigabyte). If K, M, or G is not specified, K is the default. The maximum value is 2G-1. The upper limit defaults to 2G-1 if any of the following statements is true:

- The value specified is not large enough to hold the largest primary and secondary storage allocations that are defined.
- The value specified exceeds 2G-1.
- No value is specified.

**PRDR=**

Specifies the name of the IMSRDR procedure in IMS.PROCLIB, or SYS1.PROCLIB, used in /START REGION commands. The name can be up to eight characters in length. The default is IMSRDR.

**PREINIT=**

Specifies a two-character suffix for DFSINTxx, the IMS.PROCLIB member that lists the preinitialization modules that are to receive control. For details on how to define the member, see Chapter 8, "Tailoring the IMS System to Your Environment," on page 327.

**PREMSG=**

Specifies whether, (Y) or (N), you want to receive or suppress the DFS000I prefix message that precedes all DBCTL system messages and command responses.

If the DFS000I prefix message is suppressed (PREMSG=N), all system messages are issued with the IMSID appended to the last line. The message can be a single line with a maximum of 121 characters of message text followed by XXXX, where XXXX is the IMSID. Or the message can be multiple lines with each line containing a maximum of 71 characters of message text. The IMSID will be appended to the last line of a multi-line message or will be the only text of a final line.

Display command output is prefaced by a prefix message:

```
DFS4444I DISPLAY FROM ID=XXXX
```

where XXXX is the IMSID. The message text is in columns 2-71 of all subsequent lines. Column 1 of display output contains + if the message text continues to the next line; otherwise, column 1 is blank.

If PREMSG=Y (the default value) is selected, all system messages and command responses are issued as multi-line messages.

The first line is:

```
DFS000I MESSAGE(S) FROM ID=XXXX
```

where XXXX is the IMSID. The message starts on the second line.

**PRLD=**

Specifies a two-character suffix for DFSMPLxx, the IMS.PROCLIB member identifying the modules to be preloaded in the region or partition. For more information, refer to "Member DFSMPLxx" on page 363 and "Making High-Use Program Modules Resident" on page 363.

**PSB=**

**For batch procedures:** An optional parameter specifying a PSB name when the PSB name and application program name are different.

The use of generated PSBs (GPSB) is supported in a TM batch environment. If the application program requires a GPSB, it must use the PSB parameter to indicate that request. In this case the PSB parameter does not specify the name of the PSB. Instead, it specifies a code that requests the use of a generated PSB and its language type. The specification of the code as the PSB

parameter causes the specified GPSB to be utilized for the batch application program. The application program name (MBR parameter) is used as the GPSB name. GPSBs are not available in DB batch.

The following coded character strings are used to identify the use of a GPSB and its language:

**DFS\$\$ASM** GPSB for the assembler language format.

**DFS\$\$COB** GPSB for the COBOL language format.

**DFS\$\$PLI** GPSB for the PL/I language format.

**DFS\$\$PAS** GPSB for the Pascal language format.

**Note:** The coded character strings (DFS\$\$xxx) are valid only in an IMS DCCTL environment.

**For online procedures:** Specifies the amount of subpool 231 storage to be allocated to the PSB pool. The value can be specified as one-to-six numeric characters or one-to-five numeric characters followed by K (kilobyte), M (megabyte), or G (gigabyte). If K, M, or G is not specified, K is the default. The maximum value that can be specified is 2G-1. If the value specified exceeds 2G-1 the default is 2G-1. The value specified is rounded up to the nearest page boundary. The default is 0.

The output of the ACBGEN utility indicates the maximum PSB size and the size of each PSB processed. This output should be examined before specifying the PSB= parameter.

For the FDR procedure, when LSO=S is specified, the sum of the values for the CSAPSB and DLIPSB defines the PSB pool size. If PSB is also specified, the larger value (PSB or the sum of CSAPSB and DLIPSB) is used.

#### **PSBW=**

Specifies the amount of subpool 231 storage to be allocated to the PSB work area pool. The value can be specified as one to six-numeric characters or one to five-numeric characters followed by either K (kilobyte), M (megabyte), or G (gigabyte). If K, M, or G is not specified, K is the default. The maximum value that can be specified is 2G-1. If the value specified exceeds 2G-1 the default is 2G-1. The pool size specified is rounded up to the nearest page boundary.

The output of the ACBGEN utility indicates the maximum work area size among the work area sizes required by each PSB. Depending on the execution environment, the work area size for a given PSB may be increased to the size of the long message queue buffer or to the size specified plus the maximum segment size of the segments processed by this PSB. The ACBGEN output should be examined before specifying the PSBW= parameter

This parameter is not needed for an RSR tracking subsystem.

For CSAPSB and PSBW, DFSIINS0 obtains contiguous space in the z/OS common area. If this storage is unavailable in the z/OS common area, ABENDU0717 occurs.

#### **PST=**

Specifies a one- to three-digit number of partition specification tables (PSTs) to be allocated during system initialization. Each active dependent region, DBCTL thread, or ODBA thread uses a PST. If additional dependent regions or threads are required during peak periods, IMS dynamically allocates additional PSTs as the dependent regions or threads are started. PSTs can be allocated up to the MAXPST value. During idle periods, the PSTs are released. The number of

PSTs specified on PST= is maintained. For example, if PST=15, it is assumed that approximately 15 dependent regions or threads are always active. During peak periods, if additional dependent regions or threads are started, IMS dynamically allocates a PST for each dependent region or thread, up the MAXPST value. As PSTs are terminated, the PSTs are released until a value of 15 PSTs is reached. This support is within the constraints of the operating system or subsystem being used. The default is 0.

**PWFI=**

Specifies the pseudo-wait-for-input (PWFI) parameter. N is the default.

**Y** Activates pseudo wait-for-input (PWFI). If the application program issues a Get Unique (GU) call to get a message from the IMS message queue, MODE=SINGLE is specified on the TRANSACT macro and no message is available, IMS checks for other work for this region. If no other work is available, instead of returning status QC to the application program, IMS enqueues this region on scheduler subqueue 6 indicating that it is a PWFI region. If the next message is for the transaction scheduled in this MPP region, this region is dequeued from subqueue 6 and posted. IMS then returns the new message to the application program. This eliminates the rescheduling of IMS resources.

**Related Reading:** For more information on PWFI, see *IMS Version 9: Administration Guide: System*.

**N** Disables PWFI. If the program issues a Get Unique (GU) call to get a message from the IMS message queue but no message is available, the application program receives status QC.

**QBUF= through RVFY=****QBUF=**

Specifies the one- to four-digit number of message queue buffers in subpool 0 to be allocated to the queue pool. In the shared queues environment, the number you specify for QBUF= is used as the initial number of message queue buffers allocated to the queue pool. The number of message queue buffers is dynamically expandable in the shared queues environment. The minimum number you can specify is three and the maximum is 9999. If you specify a value less than three, the number of buffers in the queue pool is set to three. If you do not specify a value, the number defaults to the value generated during system definition for the BUFFERS= parameter of the MSGQUEUE macro.

**QBUFHITH=**

Specifies a one- to three-digit number from 1 to 100 that establishes the high threshold percentage for the message queue buffer. When the buffer reaches the high threshold, it is dynamically expanded. If you do not specify a value for this parameter, the default is 80%.

**QBUFLWTH=**

Specifies a one- to three-digit number from 1 to 100 that establishes the low threshold percentage for the message queue buffer. When the message queue buffer reaches the low threshold, it is compressed. Compression only occurs when the number of message queue buffers exceeds the number allocated by system definition. If you do not specify a value for this parameter, the default is 50%. This parameter is valid only in the shared queues environment.

**QBUFMAX=**

Specifies the one- to four-digit number that indicates the maximum number of message queue buffers for the queue pool. The minimum number you can specify is 200 and the maximum is 9999. If you specify a value less than 200,

the number of buffers in the queue pool is set to 200. If you do not specify a value for this parameter, the number of buffers in the queue pool is unlimited.

This parameter is only valid in a shared queues environment. If you do not specify a value for the SHAREDQ= parameter, this parameter is ignored.

**QBUFPCTX**

Specifies a one- to three-digit number in the range of 1 to 100 that establishes the percentage of the originally allocated message queue buffers that are dynamically expanded when the limit specified by the QBUFHITH= parameter is reached. The default for this parameter is 20%.

**QBUFSZ=**

Specifies a one- to five-digit number which represents the size in bytes of the message queue buffers (in core buffers) used by the queue manager. You must specify a size greater than or equal to the length specified by LGMSGSZ= The size must be equal to or less than 30632 bytes which is the maximum length of the long message record. The size specified is rounded up to the nearest multiple of four. If you specify a value greater than 30632, the size is set to 30632. If you do not specify a value, or if you specify a value less than the length of the long message record, the size is obtained from the record length of the long message record. The DCBs for the message queue data sets are built during IMS system definition processing, even though the data sets are not used.

This parameter is only valid in the shared queues environment. If you do not specify a value for the SHAREDQ= parameter, this parameter is ignored. If you do specify a value for the SHAREDQ= parameter, the value specified in the QBUFSZ= parameter is used to determine the size of the message queue buffers.

**QTL=**

For a non-shared queues environment, specifies the lower threshold percentage (1 to 99) for each message queue data set. The default is 60%. If this threshold is crossed and if the associated exit has not been modified, a message is issued.

For a shared queues environment, specifies the lower threshold percentage (1 to 99) for the DRRN in-use count. The default value is 60%. If this threshold is crossed, and if the high threshold count was reached, a message is issued and all messages in an IWAITstate are posted.

QTU must be higher than QTL, and 0 is not valid. Either error results in the defaults being used.

**Related Reading:** For more information on message queue data sets, see: *IMS Version 9: Customization Guide*.

**QTU=**

For a non-shared queues environment, specifies the upper threshold percentage (2 through 100) for each message queue data set. The default is 75%. If this threshold is crossed, and the associated exit routine has not been modified, a message is issued.

For a shared queues environment, specifies the upper threshold percentage (2 to 100) for the DRRN in-use count. The default value is 75%. If this threshold is crossed, and if the high threshold count was reached, a message is issued and all messages being retrieved from shared queues are put in an IWAITstate until the lower threshold is reached.

**RCF=**

Specifies whether RACF is to be used for transaction authorization or signon authorization checking.

- A** Includes options T, C, and S.
- B** Includes option A, but negates the loading of the signon verification security table (DFSISSOx) from IMS.MATRIX.
- C** Specifies that RACF is to be used for ETO terminal command authorization.
- N** Specifies that no signon, transaction, or command authorization is to be performed by RACF.
- R** Includes option s, but negates the loading of the signon verification security table (DFSISSOx) from IMS.MATRIX.
- S** Specifies that RACF is to be used for static and ETO terminal command authorization. If signon verification is not explicitly specified during IMS system definition (or with the SGN= parameter), signon verification security will be set on by this specification of command authority.
- T** Specifies that RACF is to be used for signon and transaction authorization.
- Y** Includes options T and C.

If your installation has more than 65536 terminals, you must use RACF for static LTERM security.

If the RCF parameter is not specified, the default is the value specified at system definition.

**RCFTCB=**

Specifies a number between 1 and 20 indicating how many RCF TCBs are defined in your system. If you specify a null value or an invalid value, IMS uses the default of 1 RCF TCB. Increasing the number of RCF TCBs allows you to customize your system to achieve maximum parallelism and improved performance during RACF sign on and sign off calls.

**RDMNM=**

Specifies the Recovery Data Manager PROCLIB member name. If this optional parameter is not specified, the default RDM is used. If the Recovery Data Manager PROCLIB member is not found, Online Recovery Service uses default values.

**READNUM=**

Specifies the number of input devices that are to be utilized for Online Recovery Service. If input data resides on tape, READNUM specifies the maximum number of tape drives that can be concurrently allocated for Online Recovery Service. The allowable range is 1 through 99. The default value is 3.

**RECA=**

Specifies the number of receive-any buffers (1 to 500). This parameter overrides what is specified in the RECANY= parameter on the COMM macro.

If MNPS is used for XRF, the RECA= specification indicates the number of buffers for the MNPS ACB. IMS automatically allocates one additional buffer for the APPLID ACB.

**RECASZ=**

Specifies a one- to five-digit number indicating the size of the receive-any

buffers. Valid values are from 112 to 30720. If you specify a null or invalid value, IMS uses the value specified in the RECANY parameter of the COMM macro used in the IMS definition. If that value is 0, IMS uses a default value of 2100.

**RES=**

Specifies whether (Y) or not (N) the PSBs or DMBs defined in system definition macros APPLCTN or DATABASE as RESIDENT should be made resident during system initialization. The default is Y.

**REST=**

Specifies a two-digit number. A number greater than 00 indicates that the utility is to be restarted. A specification of 00 indicates no restart.

REST=*nn* specifies from which set of utility control statements starts processing. Values 00 and 01 start with the first set. Sets of utility control statements with lower values than that specified by *nn* are not processed. A set of control statements is defined by the GO statement.

**RGN=**

Specifies the region size for this execution. For Fast Path, the default is 100K.

If TLIM is specified as a number greater than one, the RGN symbolic parameter must be increased approximately 50K bytes to allow virtual storage overhead for handling RE-IN-STATE situations. Insufficient virtual storage specification results in an S106 abend of the IFP region.

**RGSUF=**

Specifies a three-character suffix for DFSPBxxx. This member contains default values for JCL EXEC statement parameters. It minimizes the number of EXEC parameter overrides needed so the EXEC parameter string does not exceed the 100 byte limit.

**RRS=**

Specifies whether (Y) or not (N) the registration and connection to RRS should be made. The default is N.

**RSRMBR=**

Specifies (for RSR systems only) the two-character suffix for an RSR member DFSSRSRxx in IMS.PROCLIB. The default is 0.

**RST=**

Specifies UCF restart. A value of 0 (no) or 1 (yes) must appear in the generated JCL statement for this parameter.

**Related Reading:** For more information on UCF restart, see *IMS Version 9: Utilities Reference: Database and Transaction Manager*.

**RVFY=**

Specifies if password reverification is activated (Y) or not (N). The default is N.

**SAV= through SYS2=****SAV=**

Specifies a one- to three-digit number of dynamic save area sets for communication terminal I/O requests. IMS expands the number of SAPs up to ten times this number as it needs to process the work being given to the system.

You can use the following formula for calculating the number of SAPs that your IMS system requires:

**Formula for calculating the number of SAPs:**



```

|      ((# of VTAM terminals + 39) ÷ 40)
|      + ((# of MSC links + 1) ÷ 2)
|      + 1 (if VTAM required) + 60 (if ETO defined)
|      + 4 (base itasks)

```

| Valid values are from 1 to 999.

| **Note:** Message DFS0769I could be issued while IMS is expanding the number  
| of SAPs. If this message is issued, see *IMS Version 9: Messages and*  
| *Codes, Volume 2* for additional information.

### SCOPE=

Specifies whether intersystem sharing is to be performed. If SCOPE=LOCAL is specified, sharing is limited to intrasystem, and XCF and SLM are neither required nor used. If SCOPE=GLOBAL or SCOPE=NODISCON is specified, intersystem sharing is performed, and both XCF and SLM are required. This parameter must always be specified, because no default value exists.

With NODISCON, there is less impact on other systems when an IMS fails because z/OS is not required to perform certain recovery actions when IRLM DISCONNECTS from the group. NODISCON can also mean that IMS restarts more quickly after an IMS terminates normally or abnormally because it does not have to wait for IRLM to rejoin the IRLM data sharing group. Under normal situations, IRLM Disconnects from the group ONLY when there are no IMS systems identified to it. NODISCON has no affect as long as there is at least one IMS identified to the IRLM.

Sharing can be performed between up to 248 IRLMs on a single OS/390 image. This can be done for pre-installation testing. In this case, SCOPE=GLOBAL must be specified for all IRLMs.

### SGN=

Specifies whether the signon verification function is active and whether the signon verification security table are to be loaded from IMS.MATRIX.

**Related Reading:** For more information, see “Establishing IMS Security” in *IMS Version 9: Administration Guide: System*.

The possible values are:

- D** Includes option F, but negates the loading of the signon verification security table (DFSISSOx) from IMS.MATRIX.
- E** Includes option G, but negates the loading of the signon verification security table (DFSISSOx) from IMS.MATRIX.
- F** Specifies that the MTO cannot negate the activation of the signon verification function or the loading of the signon verification security table from IMS.MATRIX.
- G** Includes options F and M.
- M** Specifies that a single user ID can sign on to multiple terminals. If the user structure name is different than the user ID for an ETO user (DFSSGNXO exit returned a user structure name that is not the same as the user ID), SGN=M must be specified for this user to be able to have multiple signons. SGN=M does not activate the signon verification function or cause IMS to load the signon verification security table (DFSISSOx) from IMS.MATRIX.

The SGN= parameter has ramifications when specified on the first IMS to become active in an IMS Version 9 IMSplex. The SGN= value specified on that first active IMS will be used (and enforced) for the

entire IMSplex unless overridden by a /NRE or /ERE command. For example, if the first IMS to connect to the IMSplex has single signon specified, this particular IMSplex will require all IMSs that subsequently connect to this IMSplex also have single signon specified. Similarly, if the first IMS to connect to the IMSplex has multiple signon specified, this particular IMSplex will require all IMSs that subsequently connect to this IMSplex also have multiple signon specified.

If IMSs try to join the IMSplex and they have a different value specified on the SGN= parameter, IMS issues an error message. For example, if IMSA connects to IMSplex1 first (with SGN=W) and then IMSB connects to IMSplex1 (with SGN=Z), an error message will be issued.

If all the IMSs in the IMSplex leave that IMSplex, then the first IMS to rejoin the IMSplex sets the SGN= value.

**Important:** Implementing multiple signon (SGN=M) requires a cold start of IMS.

- N** Specifies that the signon verification function is not to be activated and the signon verification security table (DFSISSOx) is not to be loaded unless overridden by the MTO. For a cold start, this implies that signon verification security is not going to be in effect.
- W** Includes option Y, but negates the loading of the signon verification security table (DFSISSOx) from IMS.MATRIX.
- X** Includes option Z, but negates the loading of the signon verification security table (DFSISSOx) from IMS.MATRIX.
- Y** Specifies that the signon verification function is to be activated and the signon verification security table (DFSISSOx) is to be loaded, unless overridden by the MTO.
- Z** Includes options Y and M.

The signon verification security table (DFSISSOx) is created by the Security Maintenance utility (SMU) when the utility processes )( SIGN STERM statements. If all of your terminals are dynamic (required to sign on) or if you have no static terminals that are required to sign on, then you do not want IMS to attempt to load the signon verification security table. You want to only specify parameters that will not cause IMS to load this table.

Parameters that will cause IMS to attempt to load this table are:

```
RCF=A | S
SGN=F | G | Y | Z
TRN=F | Y
```

If the previous specifications are used when you do not intend to load the signon verification security table, the following messages might be issued erroneously:

- DFS171A with RETURN CODE=1000 (during restart)
- DFS3436A with RETURN CODE=0408 (during online change)

#### **SHAREDQ=**

Specifies a three-character suffix for the shared queues IMS.PROCLIB member, DFSSQxxx. When you specify this parameter, IMS uses the Common Queue Server (CQS) to place messages on the shared queues that reside in a

coupling facility structure. If you do not specify this parameter, messages are placed on the message queue data sets.

There is no default for this parameter, it must be explicitly specified.

#### **SHMSGSZ=**

Specifies a one- to five-digit number that represents the length in bytes of a short message record. The length of the short message record cannot exceed 30632 bytes. The length must also be greater than or equal to the number of bytes in one and one-half times the maximum message prefix length plus four. If you specify a smaller value, the message size is set to the greater value between the size in the DCB LRECL and one and one-half times the maximum message prefix size plus four. If you specify a value greater than 30632, the size is set to 30632. If the specified value is usable, it is rounded up to the nearest multiple of four.

This parameter is only valid in the shared queues environment. If you do not specify a value for the SHAREDQ= parameter, the SHMSGSZ= parameter is ignored. If you specify a value for SHAREDQ=, but not for SHMSGSZ=, the size of the short message record is obtained from the DCB generated for the short message queue data set. The DCBs for the message queue data sets are built during IMS gen processing even though the data sets are not used.

Choose values for the SHMSGSZ= and the LGMSGSZ= based on the sizes of the messages processed by the queue manager. When determining which size message queue record to use, the queue manager calculates the size of the message prefix, subtracts that value from the size of the short message queue buffer and then doubles the remaining value. It then compares that value to the average user data length for the destination. If the remainder buffer size is greater than or equal to the average user data size, then the short message queue buffer is used. If the remainder buffer size is less than the average user data size, then the large message queue buffer is used. The length of the message prefix varies based on the IMS system options specified.

**Related Reading:** For a list of message prefix lengths see Table 22 on page 120.

#### **SOD=**

Specifies a one-character class for SYSOUT to be used for the spinoff main storage dump. If omitted or specified as 0, no spinoff dump is taken.

#### **SOUT=**

Specifies the class assigned to SYSOUT DD statements. The default is A.

#### **SPAP=**

This parameter is no longer used.

#### **SPIE=**

Specifies the SPIE option to:

- 0** Allow your SPIE, (the SPIE established by the application program), if any, to remain in effect while the application program call is being processed.
- 1** Negate your SPIE while the application program call is being processed. Negated SPIEs are reinstated before returning to the application program.

Under z/OS, if SPIE=1, a SPIE must be established; otherwise, system abend ABEND46D is issued.

A value of 0 or 1 must appear in the generated JCL statement for this parameter.

**Related Reading:** For further information about IMS and SPIEs, refer to the topic, “How PL/I-IMS Error Handling Operates” in *OS PL/I Version 2 Programming Guide, SC26-4307*.

**SPM=**

Specifies a two-character suffix for the storage pool manager PROCLIB member, DFSSPMxx.

**SRCH=**

Is the module search indicator for directed load.

- 0 Standard search
- 1 Search JPA and LPA before searching PDS

**SSM=**

Specifies a one- to four-character identifier. When building IEBUPDTE JCL, you must generate the member name by concatenating this SSM identifier to the IMSID.

All external subsystems to be accessed by any region must be defined in this member. Essentially, the member specified by the IMS procedure is a master list. Dependent region procedures (DFSMMPR, IMSBATCH, and IMSFP) can define subset lists. Each subset list can define all, some, or none of the external subsystems defined in the IMS master list. The SSM entry in each dependent region controls that region's access to subsystems.

Specifying SSM in a dependent region procedure but not in the IMS procedure is invalid, and no connection is made.

- To allow a region to access all subsystems defined in the IMS procedure, do **not** code SSM. However, if preferred, the SSM entry can specify the same member as the IMS procedure.
- To allow access to selected subsystems, specify a member defining only those subsystems a region is allowed to access. For example, if the IMS procedure member defines subsystems A, B, and C, but a region is only to access A and C, the region's SSM should specify a member defining only A and C. Thus, the SSM member is a subset of the IMS procedure member.
- To prevent this region from accessing any subsystem, specify a member containing no entries.

**STIMER=**

NOTE: The following description applies to message driven programs only.

Specifies the processor time statistics to be gathered.

- 0 No processor time statistics are to be gathered. No STIMER/TTIMER sequence is issued.
- 1 No DL/I processor time is to be included in the processor time statistics. An STIMER/TTIMER sequence is issued once for each program invocation and once for each DL/I call.
- 2 If the LSO=S region startup parameter has been selected, processor time statistics include both application program time and DL/I processing time. One STIMER/TTIMER sequence is issued for each program invocation. This is the default.

If you use the IMS timing services STIMER=1 or STIMER=2, you should ensure that your application programs do not use or invoke SVC2E or SVC2F (STIMER

or TTIMER). If these services are invoked, they negate the function of controlling application programming loops, and invalid time statistics might be gathered. Additionally, ABENDU240 might occur. You should also be aware that the LSO option you select affects the meaning of the processor time statistics. If LSO= is specified, most DL/I processing time is not included in the statistics.

**STIMER=**

NOTE: The following description applies to non-message driven programs only.

Specifies the processor time statistics to be gathered.

**0** No processor time statistics are to be gathered. No STIMER/TTIMER sequence is issued.

**1 or 2** The timer is set and processor time statistics are gathered. If the LSO=S option has been specified, the statistics include both application program time and DL/I processing time. Most DL/I processing time is not included if LSO=Y is specified. One STIMER/TTIMER sequence is issued for each program invocation.

This value should only be specified for a BMP when gathering statistics. Specifying STIMER=1 for a BMP is similar to specifying STIMER=2 for an MPP.

If no value is entered for either STIMER= or CPUTIME=, the STIMER= parameter defaults to 0 and the timer is not set. If no value is entered for CPUTIME=, the timer is not set and the STIMER= parameter defaults to 0. If a value is entered for CPUTIME=, and a value other than 0 is entered for STIMER= or STIMER= is not specified, the STIMER= parameter defaults to 2, and the timer is set.

**SUF=**

Specifies the one-character suffix for the control program name. This allows multiple copies of the IMS nucleus to reside on IMS.SDFSRESL.

This parameter cannot be changed at emergency restart.

**SVC2=**

Specifies the Type 2 SVC number that IMS will use. If the parameter is not specified, then the value specified in the IMSCTF system definition macro is used.

The SVC2 parameter overrides the value specified on the IMSCTF macro at system definition time.

**SVSODR=**

This parameter is optional. It specifies options for processing shared VSO areas at a remote site. The options take effect during an emergency restart when IMS connects to and creates a new structure for a shared VSO area.

**NONE** Specifies that processing is unchanged from existing shared VSO emergency restart processing. This is the default.

**AUTO** Specifies that during emergency restart, if IMS connects to a new structure for a shared VSO area, that area is automatically marked as recovery needed in the RECON data set. If the connection is made to a failed persistent structure, processing continues as normal.

**DRRS** This option is specified for a system that is to be restarted at the disaster recovery remote site. Typically, when an emergency restart occurs at a remote site, the connection creates a new structure and the areas need to be marked as recovery needed in the RECON data set. However, when the DRRS option is specified, IMS recognizes that the

emergency restart is at a remote site and processes accordingly. During area-open processing for a shared VSO area, IMS will not attempt to connect to shared VSO structures. The areas are marked as recovery needed in DBRC. Specifying this option saves the overhead of doing a connect and then a disconnect for each shared VSO area.

**WTOR**

Gives the user the option of marking shared VSO areas as recovery needed. During an emergency restart, IMS connects to a shared VSO structure. If the connection creates a new structure, a WTOR is generated. The prompt then gives the user the option of continuing to process with the new structure or marking the area as recovery needed. Based on the reply to the WTOR, IMS processes accordingly. A WTOR is issued for each shared VSO area that is opened during an emergency restart when the connection creates a new structure.

The WTOR message is:

```
DFS2853A A NEW STRUCTURE WAS CREATED FOR AREA AAAAAAAA.
REPLY 'C' TO CONTINUE OR 'R' TO RECOVER
```

**Note:** These options can also be specified at a local site.

**SWAP=**

Makes address space swappable (Y) or nonswappable (N). The default is Y.

If you are using DBRC, you might want to choose option N because of the possibility of being swapped out while holding the reserve on RECON.

If you are using IRLM for block-level data sharing, the SWAP parameter has no effect. If you specify IRLM=Y, a batch job is always made nonswappable to prevent the job from obtaining locks for resources and then being swapped out.

**SYS=**

Specifies an optional second-level dsname qualifier for those data sets designated as “mandatory shared” in an XRF complex. When specified, the operand must be enclosed in single quotes and must include a trailing period.

**Example:** SYS='IMSA.'

**SYS1=**

Specifies an optional second level dsname qualifier for those data sets designated as “mandatory replicate” in an XRF complex. When specified, the operand must be enclosed in single quotes and must include a trailing period.

**Example:** SYS1='IMSA.'

**SYS2=**

Specifies an optional second-level dsname qualifier for those data sets designated as “optional replicate” in an XRF complex. When specified, the operand must be enclosed in single quotes and must include a trailing period.

**Example:** SYS2='IMSA.'

**TCORACF= through YEAR4=****TCORACF=**

Specifies whether (Y) or not (N) RACF should be called to perform an authorization check of commands from a TCO script. The default is N.

Because the TCORACF specification is not included in a checkpoint record, you can change the TCORACF value each time IMS is initialized.

**TEST=**

Specifies whether (1) or not (0) the addresses in your call list should be checked for validity. A value of 0 or 1 must appear in the generated JCL statement for this parameter. An address is invalid if it is either lower than the lowest address not in the z/OS nucleus or higher than the highest address in virtual storage of the machine.

**TLIM=**

Specifies a two-digit termination limit option with a decimal number between 01 and 99. When the number of application program abends reaches this limit, the message region is automatically terminated. The accumulated number of abends for a region is incremented only if the transaction is not requeued. TLIM is not incremented in the case of a pseudoabend. This allows z/OS to print the accumulated SYSOUT data sets. If omitted or if 0 is specified, no limit on program abends is set, and the message region is not automatically terminated. For an application defined as PGMTYPE=BATCH in the APPLCTN macro statement, the optional parameter is ignored if specified. The default is 1.

**TMINAME=**

Specifies the transport manager instance name the batch job is to use. If TMINAME= is not specified in the DBBBATCH procedure, the default TMI name is the TMI name specified in the IMSCTRL macro, or blanks if no TMI name is specified.

**TRACE=**

Specifies whether the IRLM is to turn on traces. Tracing is initialized at IRLM startup. The default is TRACE=NO.

**NO** Traces are not captured unless you use the z/OS TRACE CT command.

**YES** Initiates IRLM tracing.

**TRACK=**

Specifies the level of recovery tracking for the RSR tracking subsystem.

TRACK=NO is used by the IMS active subsystem and XRF alternates subsystems, and specifies that no recovery tracking be done. This is the default, so existing procedures do not need to be changed.

TRACK=RLT is used by the RSR tracking subsystem, and specifies that recovery-level tracking be done.

TRACK=DLT is used by the RSR tracking subsystem, and specifies that database-level tracking be done.

**TRN=**

Specifies whether transaction authorization checking is to be performed. If signon verification security is not explicitly specified during IMS system definition or with the SGN= parameter, it will be set on by the specification of transaction authorization.

**E** Includes option F, but negates the loading of the signon verification security table (DFSISOx) from IMS.MATRIX.

**F** Specifies that the MTO cannot negate transaction authorization checking when issuing the /NRESTART command.

**N** Specifies that the MTO can optionally enable transaction authorization checking by specifically requesting transaction authorization checking on the /NRESTART command.

- X** Includes option Y, but negates the loading of the signon verification security table (DFSISSOx) from IMS.MATRIX.
- Y** Specifies that the MTO can optionally override transaction authorization checking by specifically requesting no transaction authorization checking on the /NRESTART command.

**TSR=**

Specifies the chosen time stamp representation.

- U** The time stamp extension to the I/O PCB contains UTC.
- L** The time stamp extension to the I/O PCB contains local time. The default is L extension to the I/O PCB. If you specify L, local time becomes the date extension to the I/O PCB.T

The default is L.

**UHASH=**

Specifies the one- to eight-character name of the Fast Path user-hashing module pointed to by the USERLIB or IMS.SDFSRESL DD statement

In a multiple-IMS configuration, the same hashing module must be used for all systems. If you do not specify this parameter, the hash module specified on the USRHASH= parameter of the FPCTRL macro in system definition is used.

**Related Reading:** For more information related to HASH modules see *IMS Version 9: Customization Guide*.

This parameter cannot be changed at emergency restart.

**UHTS=**

Specifies a one- to five-digit number of user (SPQB) hash table slots. Valid values are from 0 to 32767. The default is 256. If you specify a null or invalid value, 256 is assumed. For more information see "Specifying the Number of Hash Table Slots" on page 462.

**USERVAR=**

Specifies the user name for the IMS active subsystem. It is optional.

The *username* specified for the USERVAR= parameter for an RSR active site must remain the same in the event of a takeover and restart from a remote site. Likewise, the logical unit type 6.2 (LU 6.2) *uservars* specified at an RSR active site must remain the same in the event of a takeover and restart from a remote site.

For an XRF-capable system, the USERVAR= parameter will override the USERVAR specified in the DFSHSBxx procedure library member.

When MNPS=*name* is specified, the USERVAR specification is ignored, and IMS uses MNPS for XRF terminal switching.

**VALCK=**

Specifies the validity-check option:

- 0** Don't check the validity of addresses in the user's call list. This is the default.
- 1** Check the validity of the addresses in the user's call list.

An address is invalid if it is either lower than the lowest address not in the z/OS nucleus or higher than the highest address in virtual storage.

**VAUT=**

Specifies whether (1) or not (0) IMS is to use the VTAM authorized path facility.



**VFREE=**

Virtual Fetch is no longer supported by the operating system. This keyword, if specified, is ignored by IMS.

**VSFX=**

Virtual Fetch is no longer supported by the operating system. This keyword, if specified, is ignored by IMS.

**VSPEC=**

Specifies a two-character suffix for DFSVSMxx, the IMS.PROCLIB member that contains control statements. The default is 00. The default member in IMS.PROCLIB is DFSVSM00.

Control statements in DFSVSMxx define the following:

- Size and number of buffers in a subpool
- Number and use of local shared resource pools
- Whether sequential buffering is to be used
- Various performance and trace options
- DASD logging data set requirements
- Whether dynamic allocation for IMS batch is to be used
- Coupling facility structure names for sysplex data sharing

The VSPEC= parameter performs the same function in an IMS online system that the DFSVSAMP DD statement performs in a batch system. For more information, see “Member DFSVSMxx” on page 383.

**WADS=**

Specifies whether single (S) or dual (D) logging is to be done on the write ahead data set. The default is S.

This parameter cannot be changed at emergency restart.

**WKAP=**

Specifies the amount of subpool 231 storage to be allocated to the work area pool. The value can be specified as one- to six-numeric characters or one- to five-numeric characters followed by K (kilobyte), M (megabyte), or G (gigabyte). If K, M, or G is not specified, K is the default. The maximum value that can be specified is 2G-1. If the value specified exceeds 2G-1 the default is 2G-1. The value entered is rounded up to the nearest 4K page boundary. The default is 5K. The WKAP pool size cannot be specified at system definition.

**XPLINK=**

Specifies whether (Y) or not (N) to initialize the JVM LE enclave with XPLINK(ON). You must specify XPLINK=Y if you are running SDK 1.4.1 or above. XPLINK=N is the default.

**YEAR4=**

Specifies whether the year is displayed as four digits in the time stamp as a result of the display command. The default is YEAR4=N.

**N** Displays two digits.

**Y** Displays four digits.

---

## DD Statement Descriptions

The following list contains a description of all the DD statements that can be used by the procedures supplied for IMS.

**DFSCCTL DD**

Contains the SETR or SETO statements to specify the processing options and ranges required at scheduling time. The SETO (SET Options) statement provides the ability to set the processing options to control the execution of HSSP. The SETR (SET Range) statement restricts access to specified parts of a DEDB.

**DFSDB2AF DD**

Points to the DB2 libraries that contain modules used by RRSAP. Use this DD statement in the DFSJMP or DFSJBP procedure of a JMP or JBP that accesses DB2 databases.

The following is an example of a DFSDB2AF DD statement:

```
//DFSDB2AF DD DISP=SHR,DSN=IMS.SDFSRESL
//          DD DISP=SHR,DSN=DSNxxx.DSNLOAD
//          DD DISP=SHR,DSN=DSNyyy.DSNLOAD
```

**DFSDUPPM DD**

Is generated when a second address is specified for a primary BTAM master terminal on the system definition LINE macro. The second address is for the primary BTAM master terminal on the XRF alternate system. A DD statement named DFSDUPSM can be generated for an XRF alternate secondary master terminal.

**DFSHALDB DD**

The data set can be either a member of a PDS, a PS or instream. The data set attributes must be LRECL=80 and RECFM=FB. The contents of this data set should be the HALDB control statements that restrict processing to a single partition.

**DFSESL DD**

Points to the external subsystem libraries that contain modules used by External Subsystem Attach Facility (ESAF). Use this DD statement in the dependent region procedure of an IFP, MPP, or BMP region that accesses DB2 databases using ESAF. You can also concatenate these libraries with the JOBLIB/STEPLIB libraries in the control region instead of using this DD statement. The external subsystem libraries must follow IMS.SDFSRESL.

The following is an example of a DFSESL DD statement:

```
//DFSESL DD DISP=SHR,DSN=IMS.SDFSRESL
//          DD DISP=SHR,DSN=DSNxxx.DSNLOAD
//          DD DISP=SHR,DSN=DSNyyy.DSNLOAD
```

**DFSOLPnn DD**

Defines the primary online DASD log data set, where *nn* can be any numeric value. You can specify between 3 and 100 primary DASD data sets. The block size for the online log data sets must be a multiple of 2K bytes.

**Related Reading:** For more information, see “Allocate and Catalog IMS System Data Sets” in *IMS Version 9: Installation Volume 1: Installation Verification*.

**DFSOLSnn DD**

Defines the optional secondary online DASD log data sets, where *nn* can be any numeric value. You can specify between 3 and 100 secondary DASD data sets. The block size for the online log data sets must be a multiple of 2 K.

These statements are required only if dual logging is requested. Suffixes must be consistent with the primary log data sets.

**DFSRESLB DD**

Points to an authorized library that contains the IMS SVC modules. For IMS batch, IMS.SDFSRESL and any data set that is concatenated to it on the

DFSRESLB DD statement must be authorized through the Authorized Program Facility (APF). For more information, see “Specifying the DFSRESLB DD Statement in IMS Batch Procedures” on page 194.

### DFSSTAT DD

Defines a data set describing DB call and buffering activity during an application's execution. The reports are written when the application terminates. If you are interested in receiving //DFSSTAT reports, include a //DFSSTAT DD statement in this procedure.

**Example:** //DFSSTAT DD SYSOUT=A

**Related Reading:** For more information about interpreting //DFSSTAT reports, see *IMS Version 9: Utilities Reference: System*.

### DFSTCF DD

Defines the data set specified if the time-controlled operation function is used.

### DFSVSAMP DD

Defines the following:

- Size and number of buffers in a subpool
- Number and use of local shared resource pools
- Whether sequential buffering is to be used
- Various performance and trace options
- DASD logging data set requirements
- Whether dynamic allocation for IMS batch is to be used
- Coupling facility structure names for sysplex data sharing
- Caching option for the OSAM subpools.

**Related Reading:** For additional information on OSAM subpools, see “OSAM Subpool Definition” on page 431.

The DFSVSAMP DD statement performs the same function in an IMS batch system that the DFSVSMxx DD statement performs in an online system. In a batch system, the control statements defining the preceding options are in a data set with the ddname of DFSVSAMP; in an online system, the control statements are in a member of the IMS.PROCLIB data set with the member name of DFSVSMxx.

For more information on these control statements, see “Member DFSVSMxx” on page 383.

### DFSWADSn DD

Defines the required write-ahead data sets, where  $n$  can be a number from zero to nine. If dual logging to the WADS is requested, at least two WADS data sets must be provided. If an I/O error occurs on the current WADS, the next WADS is used.

Dual WADS logging provides backup on a read error while terminating the OLDS from the WADS. The primary and secondary WADS contain the same data. Single or dual WADS logging is determined from an execution-time parameter you specify. Regardless of whether dual logging is selected, as many as ten WADS DD statements can be included. The extra data sets serve as spares on write errors. When a write error occurs, write-ahead logging continues whether single mode or dual mode is selected. One of the spare data sets is substituted for the one with the error. When required WADSs are unavailable because of I/O errors, IMS issues a warning message and continues processing in a degraded mode, truncating OLDS buffers for LWA.

When switching to a new WADS after a write error, the current online log buffer is truncated and written to the OLDS, thereby making obsolete all data currently in the WADS.

#### **FORMATA DD and FORMATB DD**

Points to IMS.FORMATB and IMS.FORMATB, which contain online MFS definitions to be used as the format library by the online system. They are required by MFS-supported terminals.

IMS limits the number of data sets that can be concatenated for MFS format libraries, IMS.FORMATB and IMS.FORMATB, to 16. All the concatenations must have like attributes.

#### **FPTRACE DD**

Defines the destination for the Fast Path trace output. Activate the trace with the DD statement in the procedure and the following command:

```
/TRACE SET ON TABLE FAST
```

#### **IEFRDER DD**

Defines the primary system log data sets. If DASD instead of tape is used for logging, substitute the appropriate DD statements. IEFRDER and IEFRDER2 can be a combination of tape and DASD. This statement is not required if the job does not declare database-update intent.

#### **IEFRDER2 DD**

Defines the secondary system log data sets. This statement is included only when dual system log data sets are used. If DASD is used for logging instead of tape, substitute the appropriate DD statements. IEFRDER and IEFRDER2 can be a combination of tape and DASD.

#### **IMS DD**

Add an IMS DD statement for IMS.PSBLIB, concatenated with IMS.DBDLIB, if GSAM or GLOBAL databases are accessed by the batch application. The statements are:

```
//IMS DD DSN=IMS.PSBLIB,DISP=SHR
// DD DSN=IMS.DBDLIB,DISP=SHR
```

The PSB for the batch application program must be contained in the IMS.PSBLIB, and the DBDs for the GSAM or GLOBAL databases referenced by the PSB must be contained in the IMS.DBDLIB.

#### **IMSACB DD**

Points to a partitioned data set that contains the output from ACBGEN. The IMSACB DD applies to the DBBBATCH procedure. The IMSACB DD should only be used if online change is not used. To avoid user error, do not use the IMSACB DD if local online change or global online change is enabled. If local online change is enabled, use the MODSTAT DD instead. If global online change is enabled, use the OLCSTAT DD instead.

#### **IMSACBA DD and IMSACBB DD**

Point to partitioned data sets containing the output from ACBGEN. Two statements are needed for online change. If any DOPT PSBs exist for an online system, they must reside in any concatenation other than the first. If OLCSTAT is omitted or cannot be accessed, the MODSTAT DD is used, if defined. If OLCSTAT and MODSTAT are omitted or cannot be used, IMSACB DD is used.

#### **IMSIRD DD**

Defines the data set that submits the modified start-up JCL for a dependent region to the JES internal reader.

**IMSLOGR**

Defines the input log data set for extended restart. This statement is required if an application program is performing an XRST call. Any program that issues symbolic checkpoint calls must also issue the XRST call and therefore requires the IMSLOGR DD statement.

**IMSMON DD**

Describes the recording device that the IMS monitor uses. This statement is produced only if LOG=MONITOR is specified in the IMSCTF macro.

**IMSRDS DD**

Defines the first restart data set; this data set contains information required for recovery, including the checkpoint ID table needed for restarting IMS. This data set does not contain any log records.

**IMSRDS2 DD**

Defines the second restart data set. It is used only in an XRF system. This data set is identical in function to IMSRDS DD.

**IMSTFMTA DD and IMSTFMTB DD**

Point to IMS.TFORMAT, which contains the online MFS descriptors for test mode online execution. These DD statements consist of IMS.TFORMAT, concatenated in front of IMS.FORMATA and IMS.TFORMAT, concatenated in front of IMS.FORMATB. If you change MFS formats online, the two DD statements might point to this single TFORMAT data set, or the DD statements might point to two separate TFORMAT data sets.

**JAVAIN DD, JAVAOUT DD, and JAVAERR DD**

Point to HFS files that Java opens and assigns to stdin, stdout, and stderr. System.in.read() will read from the HFS file that is specified in the JAVAIN DD statement. All System.out.print() calls from the IMS Java application will be directed to the HFS file that is specified in the JAVAOUT DD statement. All System.err.print() output from the IMS Java application will be directed to the HFS file that is specified in the JAVAERR DD statement.

**Example:** The following is an example of a JAVAIN DD statement:

```
//JAVAIN DD PATH=/tmp/javain,DISP=SHR
```

**Recommendation:** IBM recommends that you specify these DD statements if you want to take advantage of the function that they offer. If you do not specify the JAVAOUT DD or the JAVAERR DD statements, System.out.print() and System.err.print() calls in the IMS Java application will not generate any output.

**JCLOUT DD**

Defines a data set where output produced by a GENJCL command is to be written.

**JCLPDS DD**

Defines a partitioned data set that contains the skeletal JCL execution and default members used by the GENJCL commands.

**LGMSG DD**

Points to the normal long message data set. If a batch only execution is planned, you do not need the message queue data sets. The Queue Manager Concurrent I/O provides a facility for the IMS customer to provide multiple "Normal" short and long message queue data sets. This facility is optional and is invoked by the IMS customer providing from one to ten DD statements for the "Normal" short and long message queues data sets.

The current implementation of the "Normal" short and long message queues allows only one DD statement for each.

**LGMSGL DD**

Points to the local long message data set. This statement is used only in an XRF complex.

**MATRIXA DD**

Points to IMS.MATRIXA, which contains the security tables created by the SMU. With this DD statement, you can bring the tables online via a /MODIFY command when IMS.MATRIXB is actively being used by the control region. At any given time, the active MATRIX data set (in use in the online system) corresponds directly to the active MODBLKS set. (They must have the same suffix.) If MODBLKSA is the active data set, security data is taken from MATRIXA. It must be authorized by APF.

**Related Reading:** For more information on IMS.MATRIXA, see *IMS Version 9: Administration Guide: System*.

**MATRIXB DD**

Points to IMS.MATRIXB, which contains the security tables created by the SMU. With this DD statement, you can bring the tables online via a /MODIFY command when IMS.MATRIXA is actively being used by the control region. At any given time, the active MATRIX data set (in use in the online system) corresponds directly to the active MODBLKS set. (They must have the same suffix.) If MODBLKSB is the active data set, security data is taken from MATRIXB. It must be authorized by APF.

**Related Reading:** For more information on IMS.MATRIXB, see *IMS Version 9: Administration Guide: System*.

**MODBLKSA DD**

Points to IMS.MODBLKSA, which contains the system definition output to be brought online via a /MODIFY command when IMS.MODBLKSB is actively being used by the control region. MODBLKSA must be authorized by APF.

**MODBLKSB DD**

Points to IMS.MODBLKSB, which contains the system definition output to be brought online via a /MODIFY command when IMS.MODBLKSA is actively being used by the control region. MODBLKSB must be authorized by APF.

**MODSTAT DD**

Points to IMS.MODSTAT, the modify status data set. The MODSTAT data set contains online change status for an IMS that has local online change enabled. The MODSTAT data set keeps track of the ddnames of libraries that are active:

- ACBLIBA or ACBLIBB
- FORMATA or FORMATB
- MODBLKSA and MATRIXA or MODBLKSB and MATRIXB

The MODSTAT DD applies to the IMS procedure and the DBBBATCH procedure. The MODSTAT DD is not needed in the IMS procedure when global online change is enabled.

If the OLCSTAT DD is omitted or the OLCSTAT data set cannot be accessed, the MODSTAT DD (if defined) determines which ddname to use. If both the OLCSTAT DD and the MODSTAT DD are omitted or the OLCSTAT and MODSTAT data sets cannot be accessed, the IMSACB DD is used.

For the IMS procedure, the ddnames to use can be ACBLIBA (or ACBLIBB), FORMATA (or FORMATB), and MODBLKSA and MATRIXA (or MODBLKSB and MATRIXB). For the DBBBATCH procedure, the ddname to use is IMSACBA or IMSACBB.

**MODSTAT2 DD**

Points to IMS.MODSTAT2, which indicates the active or inactive data sets that the IMS online system should use during initialization. This statement is used only in an XRF system.

The MODSTAT2 data set, when used for online changes, is used only for local online change. Global online change (in an IMSplex) uses the OLCSTAT data set.

**MSDBCP1 DD and MSDBCP2 DD**

Defines the first pair of MSDB checkpoint data sets. IMS alternates between the two when taking an MSDB checkpoint. During an /NRE, /ERE, or IMS procedure, the most current MSDB is determined, and the MSDBs are loaded from that data set (unless the MSDBLOAD keyword is specified during warmstart).

**MSDBCP3 DD and MSDBCP4 DD**

Defines the second pair of MSDB checkpoint data sets. Used only in an XRF environment, these data sets are identical in function to MSDBCP1 and MSDBCP2. In an XRF environment, you must specify all four data sets, because both the active and the alternate subsystem each require two. Any two of the four can contain the latest MSDB checkpoint. Although an active subsystem can select the data set containing the latest MSDB checkpoint, the alternate subsystem must select the two data sets not used by the active.

**MSDBDUMP DD**

Defines a data set that contains a dump of all MSDBs. This data set specifies where to dump the MSDBs when a /DBD DB MSDB command is entered. Successive executions of the command cause the previous contents to be overlaid.

**MSDBINIT DD**

Defines a data set that contains unloaded or reconstructed MSDBs. This data set specifies where to start the initial load of the MSDBs during all cold starts and during a normal restart if the MSDBLOAD parameter is specified for the /NRESTART command. It is produced by executing the MSDB Dump Recovery or MSDB Maintenance utility. MSDBINIT can contain one, several, or all MSDBs defined.

**OLCSTAT DD**

Points to IMS.OLCSTAT, the online change status data set. The OLCSTAT data set contains online change status for an IMSplex that has global online change enabled. The OLCSTAT data set keeps track of the ddnames of libraries that are active:

- ACBLIBA or ACBLIBB
- FORMATA or FORMATB
- MODBLKSA and MATRIXA or MODBLKSB and MATRIXB

The OLCSTAT DD applies to the DBBATCH procedure.

If the OLCSTAT DD is omitted or the OLCSTAT data set cannot be accessed, the MODSTAT DD (if defined) determines which ddname (IMSACBA or IMSACBB) to use. If both the OLCSTAT DD and the MODSTAT DD are omitted or the data sets cannot be accessed, the IMSACB DD is used.

**PRINTDD DD**

Defines the output data set for the test program, including displays of control blocks using the SNAP call. It must conform to the z/OS SNAP data set requirement.

**PROCLIB DD**

Points to IMS.PROCLIB, which contains all IMS-generated cataloged procedures, jobs, and control statements.

**QBLKS DD**

Points to the normal queue blocks message data set. If a batch-only execution is planned, you do not need the message queue data sets.

**QBLKSL DD**

Points to the local queue blocks message data set. This statement is used only in an XRF complex.

**RECON $n$  DD**

Defines the DBRC RECON data sets, where  $n$  is 1, 2, or 3. Add RECON DD statements if DBRC is being used. If dynamic allocation is being used, the DD statements are not necessary. The names of these data sets must be consistent with all subsystems that share the RECON data set.

**Related Reading:** For more information on creating a RECON data set, see *IMS Version 9: DBRC Guide and Reference*.

**SHMSG DD**

Points to the normal short message data set. If a batch-only execution is planned, you do not need the message queue data sets.

**SHMSG $L$  DD**

Points to the local short message data set. This statement is used only in an XRF complex.

**STEPLIB DD**

Points to IMS.SDFSRESL, which contains the IMS nucleus and required action modules. If STEPLIB is unauthorized because of having unauthorized libraries concatenated to IMS.SDFSRESL, a DFSRESLB DD statement must be included. The STEPLIB statement need not be authorized for IMS batch.

**SYSABEND DD**

Defines a dump data set. If both a SYSABEND DD statement and a SYSUDUMP DD statement are used, the last occurrence is used to define the dump.

**SYSHALDB DD**

Contains a report of all card images of the HALDB control statements with corresponding reasons for either accepting or rejecting the control statements.

**SYS $L$ MOD DD**

Defines the destination of the output for the linkage editor.

**SYS $P$ PRINT DD**

Defines the output data set for control messages, statistics and reports produced by a utility.

**SYS $T$ SPRT DD**

Defines the output data set for TSO Parse. The data set contains all TSO Parse error messages, such as errors in parsing the DFSRSRxx PROCLIB member. This DD statement is optional and is not generated by IMS.

**SYSUDUMP DD**

Defines a dump data set. If both a SYSABEND DD statement and a SYSUDUMP DD statement are used, the last occurrence is used to define the dump.

**SYSIN DD**

Defines the control statement input data set.



**SYSLIB DD**

Points to the libraries that contain the other modules needed for the final executable module.

**SYSLIN DD**

Defines the destination of the output from the compiler.

**SYSUT $n$  DD**

Is a temporary work data set used by the compiler, where  $n$  is 1, 2, 3, or 4. This data set must reside on DASD.



---

## Chapter 5. Establishing a DL/I Interface from COBOL or PL/I

To establish a DL/I interface from COBOL or PL/I, use either the CBLTDLI procedure or the PLITDLI procedure.

---

### CBLTDLI

The following control statements are necessary to establish a COBOL to DL/I interface. The block size of the following member must be less than or equal to 3200.

```
LIBRARY SDFSRESL(CBLTDLI)    DL/I LANGUAGE INTERFACE
LIBRARY SDFSRESL(DFHEI01)   HLPI LANGUAGE INTERFACE
LIBRARY SDFSRESL(DFHEI1)   HLPI LANGUAGE INTERFACE
ENTRY DLITCBL
```

---

### PLITDLI

The following control statements are necessary to establish a PL/I to DL/I interface. The blocksize of the following member must be less than or equal to 3200.

```
LIBRARY SDFSRESL(PLITDLI)   DL/I LANGUAGE INTERFACE
LIBRARY SDFSRESL(DFHEI01)   HLPI LANGUAGE INTERFACE
LIBRARY SDFSRESL(DFHEI1)   HLPI LANGUAGE INTERFACE
ENTRY PLICALLA
```

PLITDLI is valid when using the PL/I Optimizing Compiler.



## Chapter 6. Installing and Tailoring the Interactive Dump Formatter

### In this Chapter:

- “Installing the Interactive Dump Formatter.”
- “Tailoring Your Dump Formatting Options.”
- “IRLM Use of SDUMP” on page 321.

## Installing the Interactive Dump Formatter

**Prerequisite:** Before you begin installing the Interactive Dump Formatter, be sure you have IPCS already functioning with ISPF/PDF.

1. Update the DD concatenations listed in Table 32:

Table 32. DD Concatenations to Update

DDNAME	Data Set to Be Added	Contents of Data Set
SYSPROC	IMS.SDFSCLST	CLISTS
	IMS.SDFSEXEC	REXX EXEC
ISPMLIB	IMS.SDFSMLIB	Messages
ISPLLIB	IMS.SDFSPLIB	Panels
ISPTLIB	IMS.SDFSTLIB	Tables
IPCSPARM	IMS.SDFSMAC	All IMS macros
		<b>Note:</b> DFSIPCSP is the only member that is used from this data set.
TASKLIB	IMS.SDFSRESL	Formatting modules
		<b>Note:</b> The TASKLIB concatenation is specified as part of the IPCS command invocation.

2. Start IPCS and select the dump you wish to browse.
3. From the IPCS Primary Option Menu, select Option 2.6, ANALYSIS.COMPONENT.
4. From the Component Analysis Panel, select DFSAAMPR.
5. From the IMS Dump Formatting Primary Option Menu, select Option 0, INIT.
6. From the IMS Dump Contents Status Panel, enter the job name of either the IMS control/batch region or the IMS DLISAS region.
7. Press the Enter key and then PF3 to return to the IMS Dump Formatting Primary Option Menu.
8. Select an IMS Analysis Option from the Primary Option Menu. HILEVEL is a good starting point because it provides a broad diagnostic overview of the problem.

## Tailoring Your Dump Formatting Options

The dumping options available to you include the following:

- Offline IMS or z/OS formatting from a SYS1.DUMPxx (SDUMP) data set
- Offline IMS or z/OS formatting from a SYSMDUMP data set

- Online IMS or z/OS formatting directed to either a SYSABEND or SYSUDUMP SYSOUT class
- Online z/OS formatting directed to either a SYSABEND or SYSUDUMP spinoff dump SYSOUT class

The dumping options in effect in an IMS environment depend on:

- Whether execution is online or batch.
- Whether the error might terminate IMS.
- The FMTO EXEC parameter specification.
- The SOD EXEC parameter specification (online only).
- The choice of a SYSABEND, SYSUDUMP, or SYSMDUMP DD statement.
- The z/OS dump options chosen for SDUMP, SYSABEND, SYSUDUMP, and SYSMDUMP.
- Whether the Dump Override Table contains any entries.

**Related Reading:** See *IMS Version 9: Customization Guide* for further information about Dump Override Tables.

## z/OS Dump Options

In order for IMS dumps to be useful in diagnosing problems, the proper dump options must be used (by specification or default) for each type of z/OS dump used for IMS.

The z/OS operator command DISPLAY DUMP, OPTIONS can be used to determine the dump options currently in effect on your system.

If the current dump options do not include the options listed in this section, the defaults obtained from SYS1.PARMLIB (members IEAABD00, IEADMP00, and IEADMR00) must be altered. The z/OS operator command CHNGDUMP can be used to change dump options.

- SDUMP options  
SDUMP=(CSA,PSA,RGN,SQA,SUM,TRT,GRSQ)
- SYSABEND options  
SYSABEND,SDATA=(CB,DM,ENQ,ERR,IO,LSQA,SUM,TRT)  
SYSABEND,PDATA=(JPA,LPA,PSW,REGS,SA,SPLS)
- SYSUDUMP options  
SYSUDUMP,SDATA=(CB,ERR,SUM)  
SYSUDUMP,PDATA=(JPA,LPA,PSW,REGS,SA,SPLS)
- SYSMDUMP options (**ONLINE** system only)  
SYSMDUMP=(CSA,LSQA,RGN,SQA,SUM,SWA,TRT)
- SYSMDUMP options (**BATCH** system only)  
SYSMDUMP=(LSQA,RGN,SQA,SUM,SWA,TRT)

## Online System Dumps

In the online IMS environment, the following dumps are possible:

- SDUMP to a SYS1.DUMPxx data set. (SDUMP can then be formatted using the Offline Dump Formatter utility.)  
This dump can be generated both for errors that cause IMS to terminate (*terminating errors*) and for errors that do not cause IMS to terminate (*nonterminating errors*).
- SYSABEND or SYSUDUMP (with IMS or z/OS formatting performed online).

This dump is generated for terminating errors only if SDUMP fails or was not requested.

- SYSMDUMP to a data set. (SYSMDUMP can then be formatted using the Offline Dump Formatting utility.)

This dump is generated for terminating errors only if SDUMP fails or was not requested.

- Spinoff dump to SYSABEND or SYSUDUMP (with z/OS formatting performed).

This dump is generated for nonterminating errors and can be generated in addition to SDUMP.

The IMS dumping options are controlled by the Dump Override Table entries, the FMTO EXEC parameter specification for the IMS control region, and the presence of a SYSABEND, SYSUDUMP, or SYSMDUMP DD statement. (These DD statements are mutually exclusive; provide only one.)

**Related Reading:** The Dump Override Table (DFSFDOT0) and its relationship with the various dumps are explained in the *IMS Version 9: Customization Guide*.

The FMTO options and their effect on the dumps produced are summarized in Table 33. In Table 33, the following codes are used:

- TE** Error would terminate IMS.
- NT** Error would **not** terminate IMS.
- FDDL** Formatted dump delete list processing.
- Y** Yes
- S** Yes, but only if SDUMP fails.
- N** No
- Not applicable, or not attempted.

Usage notes follow Table 33.

Table 33. FMTO Options and their effect on dumps produced.

FMTO Value	IMS Offline Formatting SDUMP		IMS Online Formatting SYSABEND SYSUDUMP			z/OS Online Formatting SYSABEND SYSUDUMP		IMS Offline Formatting SYSMDUMP		IMS Spinoff Dump SYSABEND SYSUDUMP	
	TE	NT	TE	NT	FDDL	TE	NT	TE	NT	TE	NT
D	Y	Y	S	—	S	S	—	S	—	—	Y
X	Y	N	S	—	S	S	—	S	—	—	Y
M	Y	Y	N	—	N	S	—	S	—	—	Y
R	Y	N	N	—	N	S	—	S	—	—	Y
T	N	Y	Y	—	Y	Y	—	Y	—	—	Y
P	N	Y	Y	—	N	Y	—	Y	—	—	Y
F	N	N	Y	—	N	Y	—	Y	—	—	Y
N	N	Y	N	—	N	Y	—	Y	—	—	Y
Z	N	N	N	—	N	Y	—	Y	—	—	Y

**Notes to Table 33:**

- The default is FMTO=D.
- SYSABEND, SYSUDUMP, and SYSMDUMP are mutually exclusive.
- z/OS formatting always accompanies IMS online formatting.
- Spinoff dumps can occur in addition to SDUMPs.

When using Table 33 on page 317, it is important to remember the following:

- If SDUMP is requested by the FMTO option, it is attempted first.  
If SDUMP is successful, no other dump is generated. (Spinoff dumps are an exception.)
- If SDUMP fails, if IMS online formatting is requested, and if a SYSABEND or SYSUDUMP DD statement is present, a dump with IMS and z/OS online formatting is generated for terminating errors.  
The formatted dump delete list (FDDL) consists of IMS module and control block names, and a dump or delete action indicator. For a module, the delete process removes storage occupied by the module. This causes the module to be omitted from the storage image portion of the dump. However, the module/save ID is printed in the dump. For a control block, the delete process similarly removes storage and storage image output if the block was successfully formatted.
- If SDUMP fails, IMS online formatting is not requested, and a SYSABEND or SYSUDUMP DD statement is present, a dump with z/OS online formatting is generated for terminating errors.
- If SDUMP fails and a SYSMDUMP DD statement is present, a machine readable dump is generated for terminating errors.
- If a spinoff dump SYSOUT class is specified (SOD EXEC parameter in the IMS control region), a spinoff dump is generated for nonterminating errors.

If SYSMDUMP is used, some additional considerations are necessary:

- SYSMDUMP DD statements should be present in the IMS, DBRC, and DLISAS procedures.
- The SYSMDUMP DD statement must specify DISP=MOD, because it is possible for multiple tasks to generate dumps during the processing of an error.
- The SYSMDUMP data set must be refreshed (for example, scratched and reallocated) before it can be reused. If this is not done, new dumps are appended to the end of the data set and may not be accessible to the Offline Dump Formatting utility.

Following are some additional considerations if spinoff dumps are used:

- Spinoff dumps allow the dump data set to be released to JES for immediate printing. (This can be useful when external subsystems are supported.)
- Spinoff dumps are only generated for those tasks that do not terminate IMS.
- If a SYSABEND or SYSUDUMP DD statement is not provided, the first dump is not printed.

For subsequent dumps, the dump data set is dynamically allocated using the spinoff dump class.

- The spinoff dump class (SOD execute parameter) should be the same as the SYSABEND or SYSUDUMP SYSOUT class.

If the classes are different, the first dump goes to the class specified on the SYSABEND or SYSUDUMP DD statement. The dump data set is then dynamically deallocated to free the dump for printing and then dynamically reallocated using the spinoff dump class. Therefore, all subsequent dumps go to the spinoff dump class.



- The only valid SYSABEND or SYSUDUMP DD statement parameter used for spinoff dump is the SYSOUT class. Other parameters are ignored.
- Spinoff dumps use the default output limit established during JES definition. Incomplete spinoff dumps can result if the limit specified is too low.
- SYSMDUMP should not be used in place of SYSABEND or SYSUDUMP.

## Online System Dependent Region Dumps

In the online IMS environment, the following dumps are useful for MPP, IFP, and BMP dependent regions:

- SYSABEND or SYSUDUMP (with z/OS formatting performed online).
- Spinoff dump to SYSABEND or SYSUDUMP (with z/OS formatting). This dump is available only for MPP and IFP dependent regions.
- SDUMP to a SYS1.DUMPxx data set. (SDUMP can then be formatted using the Offline Dump Formatter utility.) An SDUMP is produced in the dependent region for certain IMS system errors involving DL/I or Fast Path. In this instance, the SDUMP is controlled by the FMTO specification for the control region.

Except for the SDUMPs, the dumping options for the dependent regions are controlled by the following:

- The spinoff dump SYSOUT class (SOD EXEC parameter in the DFSMPR and IMSFP procedures)
- The presence or absence of a SYSABEND or SYSUDUMP DD statement in the dependent region JCL. (These DD statements are mutually exclusive; provide only one.)

SYSMDUMP should not be used in IMS dependent regions.

If spinoff dumps are used, some additional considerations are necessary:

- Spinoff dumps allow the dump data set to be released to JES for immediate printing. (This can be useful for “never-ending” MPP or IFP regions.)
- If a SYSABEND or SYSUDUMP DD statement is not provided, the first dump is not be printed. For subsequent dumps the dump data set is dynamically allocated using the spinoff dump class.
- The spinoff dump class (SOD EXEC parameter) should be the same as the SYSABEND or SYSUDUMP SYSOUT class.

If the classes are different, the first dump goes to the class specified on the SYSABEND or SYSUDUMP DD statement. The dump data set is then dynamically deallocated to free the dump for printing and then dynamically reallocated using the spinoff dump class. Therefore, all subsequent dumps go to the spinoff dump class.

- The only valid SYSABEND or SYSUDUMP DD statement parameter used for a spinoff dump is the SYSOUT class. Other parameters are ignored after the first dump.
- Spinoff dumps use the default output limit established during JES definition. Incomplete spinoff dumps can result if the limit specified is too low.
- If the COBOL DEBUG option is used, dumps cannot be generated.

## Batch System Dumps

In the batch IMS environment, the following dumps are possible:

- SYSMDUMP to a data set. (SYSMDUMP can then be formatted using the Offline Dump Formatting utility.) This dump is only generated for terminating errors.

- SYSABEND or SYSUDUMP (with IMS or z/OS formatting performed online). This dump is only generated for terminating errors.

The IMS dumping options are controlled by the Dump Override Table entries, the FMTO EXEC parameter specification for the IMS batch region, and the presence of a SYSABEND, SYSUDUMP, or SYSMDUMP DD statement. (These DD statements are mutually exclusive; provide only one.)

**Related Reading:** The Dump Override Table (DFSFDOTO) and how its relationship with the various dumps is explained in the *IMS Version 9: Customization Guide*. See information about Interactive Dump Formatter in the *IMS Version 9: Diagnosis Guide and Reference* for information on dialog support for offline dump formatter requests.

The FMTO options and their effect on the dumps produced are summarized in Table 34.

When using this figure, it is important to remember the following:

- If a SYSMDUMP DD statement is present, a machine readable dump is generated for terminating errors.
- If IMS online formatting is requested and a SYSABEND or SYSUDUMP DD statement is present, an IMS online formatted dump is generated for terminating errors.
- If IMS online formatting is not requested and a SYSABEND or SYSUDUMP DD statement is present, an z/OS online formatted dump is generated for terminating errors.

In Table 34, the following codes are used:

- TE** Error would terminate IMS.
- NT** Error would **not** terminate IMS.
- FDDL** Formatted dump delete list processing.
- Y** Yes
- N** No
- Not applicable, or not attempted.

Usage notes follow Table 34.

Table 34. Batch FMTO Options

FMTO Value	IMS Online Formatting SYSABEND SYSUDUMP			z/OS Online Formatting SYSABEND SYSUDUMP		IMS Offline Formatting SYSMDUMP	
	TE	NT	FDDL	TE	NT	TE	NT
D	Y	—	Y	Y	—	Y	—
X	Y	—	Y	Y	—	Y	—
M	N	—	N	Y	—	Y	—
R	N	—	N	Y	—	Y	—
T	Y	—	Y	Y	—	Y	—
P	Y	—	N	Y	—	Y	—

Table 34. Batch FMTO Options (continued)

FMTO Value	IMS Online Formatting SYSABEND SYSUDUMP			z/OS Online Formatting SYSABEND SYSUDUMP		IMS Offline Formatting SYSMDUMP	
F	Y	—	N	Y	—	Y	—
N	N	—	N	Y	—	Y	—
Z	N	—	N	Y	—	Y	—

**Notes for Table 34:**

- The default is FMTO=D.
- SYSABEND, SYSUDUMP, and SYSMDUMP are mutually exclusive.
- z/OS formatting always accompanies IMS online formatting.

Following are some additional considerations if SYSMDUMP is used:

- The SYSMDUMP DD statement should specify DISP=MOD.
- If DISP=MOD is used, then the SYSMDUMP data set must be refreshed (for example, scratched and reallocated) before it can be reused. If this is not done, new dumps are appended to the end of the data set and might not be accessible to the Offline Dump Formatting utility.

---

## IRLM Use of SDUMP

The IRLM uses the z/OS SDUMP program for dumping whenever its ESTAE or FRR routines are entered. SDUMP dumps are directed to the SYS1.DUMPxx data sets and printed using the IPCS service aid. SDUMP has the same advantages over the SYSABEND dump program as the spinoff dump program. The operator is told which data set contains the SDUMP by message IEA911E COMPLETE DUMP ON SYS1.DUMPxx.

Following are sample control statements for executing the IPCS program.

```
//SYSIN DD *
        NEWDUMP DD=INPUT
        FORMAT
        LOGDATA
        VTAMMAP
        IRLM ir1m.subsystem.name
        PRINT JOBNAME=(ir1m.job.name)
        END
```

**Related Reading:** For a description of the IPCS utility, see *MVS/ESA Interactive Problem Control System (IPCS) Users Guide*.

When running the IPCS program, you should provide a JOBLIB OR STEPLIB DD statement pointing to the library with the IRLM PRDUMP formatting routine (DXRRLM50). The IRLM control statement invokes this formatting routine to format IRLM control blocks.



---

## Chapter 7. Installing the Transport Manager Subsystem

To install the transport manager subsystem, place the initial command stream into a data set referenced by the transport manager's SYSIN DD statement. This allows you to have the component start be an automated process. The JCL procedure must be tailored to your local conventions and placed in a library accessible by the z/OS START command. Figure 29 provides an example of job control statements for the transport manager.

```
//TMS      PROC SYS2=,CMD=TMSTART
//          EXEC PGM=DFSMVCR0,PARM=(ELX[, 'IMSPLEX=cccc'])
//STEPLIB DD DISP=SHR,DSN=IMS.&SYS2.SDFSRESL
//SYSIN   DD DISP=SHR,DSN=IMS.&SYS2.PROCLIB(&CMD)
//SYSOUT  DD SYSOUT=A
//SYSPRT DD SYSOUT=A
//SYSABEND DD SYSOUT=A
//*
```

Figure 29. Transport Manager Job Control Statements

**Related Reading:** For a description of the IMSPLEX parameter, see the list of descriptions under “Parameter Descriptions for IMS Supplied Procedures” on page 255.

The transport manager subsystem requires no offline definition. All TMS definition can be performed when you start the TMS, or while the TMS is running, using transport manager commands. However, some VTAM definition is required before you can use the transport manager subsystem.

You cannot name the TMS the same name as the TM start-up member in SYS1.PROCLIB.

---

### VTAM Definition for the Transport Manager Subsystem

In order for the transport manager subsystem to use VTAM, certain VTAM definition work is required. Because the VTAM publications describe VTAM definition specifics, the example provided in Figure 30 on page 324 shows the relationships between products and indicates which pieces of VTAM definition are important.

**Related Reading:** See *IMS Version 9: Administration Guide: Transaction Manager* for related information.

```

*****
* SYS1.VTAMLST APPL definition for Transport Manager *
* Corresponds with a Transport Manager command of: *
* SET APPLID(TMP2) APPLCOUNT(9) *
* *
*****
          VBUILD TYPE=APPL
TMP2001  APPL  EAS=64,PARSESS=YES,SRBEXIT=YES,          C
          APPC=YES,AUTOSES=1,DMINWNL=32,DMINWNR=32,DSESLIM=64,  C
          MAXPVT=1024K,VPACING=10,VTAMFRR=NO,          C
          AUTH=(VPACE),MODETAB=MODELUTM,DLOGMOD=TMDEFLT
TMP2002  APPL  EAS=64,PARSESS=YES,SRBEXIT=YES,          C
          APPC=YES,AUTOSES=1,DMINWNL=32,DMINWNR=32,DSESLIM=64,  C
          MAXPVT=1024K,VPACING=10,VTAMFRR=NO,          C
          AUTH=(VPACE),MODETAB=MODELUTM,DLOGMOD=TMDEFLT
          .
          .
          .
TMP2009  APPL  EAS=64,PARSESS=YES,SRBEXIT=YES,          C
          APPC=YES,AUTOSES=1,DMINWNL=32,DMINWNR=32,DSESLIM=64,  C
          MAXPVT=1024K,VPACING=10,VTAMFRR=NO,          C
          AUTH=(VPACE),MODETAB=MODELUTM,DLOGMOD=TMDEFLT

```

Figure 30. Sample VTAM APPL Definition

**APPL**

In SYS1.VTAMLST, a group of application program minor nodes must be defined for use by transport manager and IMS systems. An example is shown in Figure 30. This definition is performed with the APPL definition statement. The “name” on the APPL statement corresponds to the transport manager SET command APPLID (VTAM application id) name. Each APPL statement must have three numeric characters (beginning with 001) appended to the one to five character name specified in the SET APPLID command. The number of APPL names defined must exceed by at least one the maximum number of IMS subsystems and ILS instances expected to execute concurrently on this CPC.

Other key specifications for APPL are:

- AUTH=VPACE to allow session pacing.
- MODETAB must specify the member name of a logon mode table in SYS1.VTAMLIB that contains entries appropriate for transport manager.
- EAS specifies the number of VTAM sessions that each ACB can have. It is defined as follows:

$$2 \times (\text{maximum number of concurrent active IMS subsystems} + \text{maximum specification of the MAXCONV value}) + 2$$

A value of 64 should be more than enough.

- ENCR=NONE should be specified unless you want VTAM encryption.
- MAXPVT=1024 allows VTAM to use one megabyte of transport manager or IMS private area storage.
- PARSESS=YES allows parallel sessions and is required.
- PRTCT correlates with “vtam-acb-password” on the transport manager SET PASSWORD command.
- SRBEXIT=YES must be specified.
- VPACING=n (start with 10) to limit VTAM buffer usage while allowing reasonable bandwidth.
- VTAMFRR=NO must be specified.

**MODEENT**

Defines an entry in a mode table when combined with MODETAB and MODEEND macros, assembled and link edited into SYS1.VTAMLIB. An example is shown in Figure 30 on page 324. For the transport manager subsystem, the primary value is to obtain a class of service for a conversation, thus choosing a network route.

- LOGMODE specifies the logon mode table entry name, correlating to DLOGMOD on the VTAM APPL definition or to MODENAME specified by a transport manager component
- COS specifies the name of an entry in the class of service table
- SRCVPAC should be zero so that APPL VPACING= will control pacing on the PLU-SLU sessions
- SSNDPAC should be non-zero so that APPL VPACING= will control pacing on the SLU-PLU sessions
- TMDEFLT must be defined as the default logon mode table entry name

**COS**

Defines an entry in a class of service table when combined with COSTAB and COSEND macros, assembled and link edited into SYS1.VTAMLIB. The recommended transmission priority to specify with virtual routes is 2.

If you have enough hardware connectivity between sites, use multiple MODEENT entries and more than one class of service to select different virtual routes, thus spreading different transport manager conversations among the hardware.

COS could be used to avoid satellite links (because of latency), particularly for LOGGER conversations.

**IOBUF**

Defines the size and number of VTAM I/O buffers. Both the size and number have a significant impact on processor consumption and effective bandwidth. 1 KB buffers provide much better performance than small buffers, but will require more storage than small buffers where network traffic consists of many small messages. The average RU size for logger to log router conversations will be close to the OLDS blocksize (typically 22 KB).

Figure 31 on page 326 provides a sample VTAM mode table definition for the transport manager subsystem.

```

*****
*  MODULE NAME:  MODELUTM                               *
*                                                       *
*  DESCRIPTIVE NAME:                                   *
*    LOGON MODE TABLE FOR Remote Site Recovery(RSR) TRANSPORT MANAGER *
*                                                       *
*  FUNCTION:  THIS TABLE IS USED BY VTAM TO DETERMINE THE BIND *
*    PARAMETERS FOR A SESSION.                          *
*                                                       *
*  NOTES:                                             *
*    SET PACING TO 10 IN DEFAULT ENTRY.                *
*    SET SSNDPAC NON-0, SRCVPAC 0,                    *
*    THIS CAUSES APPL VPACING TO BE USED.              *
*                                                       *
*****
MODELUTM  MODETAB
*****
*  DEFAULT - SEND PACING OF -0 FORCES USE OF APPL VPACING *
*****
      MODEENT LOGMODE=TMDEFLT,FMPROF=X'13',TSPROF=X'07',           X
              SSNDPAC=X'3F',SRCVPAC=X'00', TUNE ADAPT SESSION PACING X
              RUSIZES=X'8D8D', 8 * 2**D = 8*8192=65536           X
              PSNDPAC=X'00', TUNE ADAPT SESSION PACING           X
              TYPE=0, NEGOTIABLE BIND                             X
              PSERVIC=X'0602000000000000000000000000300',       X
              PRIPROT=X'B0',SECPROT=X'B0',COMPROT=X'D0B1'
      TITLE 'SNASVCMG' *@R495812*
*****
*  LOGMODE TABLE ENTRY FOR SPECIAL VTAM APPC SESSIONS THAT *
*  SUPPORT CNOS. TAKEN FROM SYS1.SAMPLIB(ISTINCLM)          *
*  @R495812*
*****
SNASVCMG MODEENT LOGMODE=SNASVCMG,FMPROF=X'13',TSPROF=X'07',           X
              PRIPROT=X'B0',SECPROT=X'B0',COMPROT=X'D0B1',       X
              RUSIZES=X'8585',ENCR=B'0000',                      X
              PSERVIC=X'0602000000000000000000000000300'        *@R495812*
      MODEEND
      END

```

Figure 31. Sample VTAM MODETAB Definition



## Chapter 8. Tailoring the IMS System to Your Environment

### In this Chapter:

- “Tailoring the IMS Procedure Library” on page 327
- “Specifying IMS PROCLIB Members” on page 333
- “Specifying IMS Execution Parameters” on page 415
- “Specifying IMS Buffer Pools” on page 430
- “Making IMS and IMSRDR Procedures Accessible to z/OS” on page 433
- “Organizing PL/I Modules for Use with the PL/I Optimizer” on page 433
- “Coding COBOL Programs That Run under IMS” on page 434
- “Specifying Sequential Buffering Control Statements” on page 434
- “Supporting CCTL Users with DBCTL Databases” on page 446
- “Managing an IMSplex Environment with a Common Service Layer” on page 450
- “Enabling ETO Support for ACF/VTAM Terminals” on page 450
- “HALDB Single Partition Processing” on page 463

### Tailoring the IMS Procedure Library

The results of Stage 1 system definition include a set of updates to IMS.PROCLIB which are listed in the Stage 1 output. Stage 2 applies these updates to the library.

You can tailor your contents in any of the following ways:

- Tailor the contents of the members before implementing Step 2 of the tailoring process or directly alter the contents of IMS.PROCLIB. See “Altering the IMS Procedure Library” on page 329 for more information.
- Use the IMS Syntax Checker. See “IMS Syntax Checker” on page 332

Remember that you can set up default values for the parameters, but you can also specify them again at region startup with the symbolic parameters. Initially, these parameters use values preset by system definition, but individual region control or tuning recommendations can override the initial or default values.

### Procedures Generated by System Definition

The generated members of IMS.PROCLIB fall into several categories depending on their use, as shown in Table 35.

Table 35. Categories of Generated IMS Members

Category	Member	Usage
Online Operation	DBC	Execute DBCTL region
	DBRC	Execute DBRC address space
	DCC	Execute DCCTL region
	DFSJBP	Execute Java non-message driven batch address space
	DFSJMP	Execute Java message program address space
	DFSMPR	Execute message processing region
	DFSWTnnn	Print IMS spool data sets
	DLISAS	Execute DL/I address space
	DXRJPROC	Execute IRLM region

Table 35. Categories of Generated IMS Members (continued)

Category	Member	Usage
	FPUTIL	Execute DEDB utilities online
	IMS	Execute control region
	IMSBATCH	Execute batch message region
	IMSFP	Execute Fast Path dependent region
	IMSMMSG	Execute message region
	IMSRDR	Read IMSMSG job
	IMSWTnnn	Execute DFSWTnnn
Online Initialization	DFSCGxxx	Specify CSL parameters
	DFSDCxxx	Specify data communication options
	DFSDRFnn	List for defining DREF requirements
	DFSFDRxx	Specify FDBR region execution parameters
	DFSFIXnn	List for page fixing blocks and modules
	DFSHSBxx	Specify XRF options used by the active and alternate subsystems in XRF complex
	DFSINTxx	Identify preinitialization modules
	DFSMP Lxx	Define preload program list
	DFSORSxx	Specify system-related startup parameters for the recovery manager
	DFSPBDBC	Specify DBCTL execution parameters
	DFSPBDCC	Specify DCCTL execution parameters
	DFSPBIMS	Specify DB/DC execution parameters
	DFSRSRxx	Specify Remote Site Recovery execution parameters
	DFSSPMxx	Specify storage pool management parameters
	DFSSQxxx	Specify shared message queue and CQS address space parameters
	DFSVSMxx	Define buffer pools, logging attributes, and other options
	DFSYDT	Specify OTMA descriptors
	DFS62DT	Specify LU 6.2 device descriptors
Online Preparation and Maintenance	ACBGEN	Maintain ACBLIB
	IMSDALOC	Maintain dynamic allocation
	IMSM SV	Offline utility for MSC
	MFSDCT	Maintain MFSDCT and default format
	MFSRVC	Maintain MFS service and index
	MFSUTL	Maintain MFS library Other procedures are: MFSBTCH1/2, MFSTEST, MFSBACK, and MFSREST
	OLCUTL	Prepare IMS for online change
	SECURITY	Execute Security Maintenance utility
Batch Operation	DBBBATCH	Execute stand-alone DL/I batch processing region
	DLIBATCH	Execute stand-alone DL/I batch processing region with block building

Table 35. Categories of Generated IMS Members (continued)

Category	Member	Usage
Alternative Batch Execution	IMSCOBGO	Compile, bind, and execute (under batch) a COBOL program
	IMSPLOGO	Compile, bind, and execute (under batch) a PL/I program
Database Definition and Access	DBDGEN	Maintain DBDLIB
	DBFMSDBx	Specify MSDBs
	FPUTIL	Execute online DEDB utilities
	MFDBDUMP	Program to list sample databases
	MFDBLOAD	Batch execution of a program to load the sample databases
Application Program Preparation	CBLTDLI	Bind control statements for COBOL
	DFSJVMAP	Map IMS Java application name to OMVS path name for Java class file
	IMSCOBOL	Compile and bind COBOL program
	IMSPLI	Compile and bind PL/I program
	PLITDLI	Bind control statements for PL/I
	PSBGEN	Maintain PSBLIB

When you perform an IMS system definition, IMS generates members of IMS.PROCLIB if PROCLIB=YES is specified (or defaulted to) on the IMSGEN macro and you are generating a CTLBLKS, NUCLEUS, ON-LINE, or ALL type of definition. If you have made any changes to a prior generated library that you want to save, specify PROCLIB=NO.

If you perform an ON-LINE, CTLBLKS, or NUCLEUS definition and have made changes affecting line groups, unit addresses, or spool output data sets, specify PROCLIB=IMS on the IMSGEN macro to replace only the member IMS.

## Altering the IMS Procedure Library

Many of the members generated in IMS.PROCLIB require alteration before they can be used in direct execution of the online system, yet they provide a convenient start to the task of defining execution JCL.

You should carefully examine the members generated as a result of your system definition.

### Renaming PROCLIB Members

Rename the IMS.PROCLIB members for your installation's requirements. The names can follow a convention that suggests ownership by a particular application system or a convention that has an implied sequence. For example:

<b>IMSCTL</b>	Control region startup (IMS renamed)
<b>IMSMSG1</b>	Message region startup (IMSMSG)
<b>IMSMSG2</b>	Second message region when required (IMSMSG)
<b>IMSWT000</b>	Spool output print procedure (named by IMS)

The system operator or MTO uses the procedure names to invoke the z/OS job execution.

To develop these members, you need to either rename them in IMS.PROCLIB or create new ones. You can add the procedures to SYS1.PROCLIB. One option of the NODE keyword on the IMSGEN macro allows you to substitute an alternative library for the members.

### Initializing Your Procedure Library

You must adjust the JCL content of the members of the procedure library. The updates you apply follow naming conventions for your IMS systems and the required DD statements.

The checklist shown in Table 36 summarizes the tailoring actions, excluding arrangements for batch executions. In all procedures, you might want to add JCL comment statements for additional documentation. Your procedure library probably does not contain all the members listed in Table 36.

*Table 36. Tailoring Actions for PROCLIB Members for Batch Executions*

<b>PROCLIB Member</b>	<b>Tailoring Action</b>
IMS	<ul style="list-style-type: none"> <li>• Checks and overrides the EXEC PARM parameter.</li> <li>• Specifies region size again.</li> <li>• Moves to SYS1.PROCLIB.</li> <li>• Adds database DD statements, if not dynamically allocated.</li> <li>• Specifies logging data sets again; allows for dynamic allocation.</li> </ul>
IMSRDR	<ul style="list-style-type: none"> <li>• Adjusts all symbolic parameters for message region start by system operator.</li> <li>• Moves to SYS1.PROCLIB.</li> </ul>
DBC	<ul style="list-style-type: none"> <li>• Checks and overrides EXEC PARM parameter.</li> <li>• Specifies region size again.</li> <li>• Moves to SYS1.PROCLIB.</li> <li>• Adds database DD statements, if not dynamically allocated.</li> <li>• Specifies logging data sets again, allow for dynamic allocation.</li> </ul>
DBRC	<ul style="list-style-type: none"> <li>• Adds RECON DD statements, if not dynamically allocated.</li> <li>• Renames the procedure, if desired, and specifies this name with the DBRCNM parameter of the IMS procedure.</li> <li>• Adjusts procedures JCLOUT and JCLPDS.</li> <li>• Moves to SYS1.PROCLIB.</li> </ul>
DFSDCxxx	Specifies data communication options.
DFSFDRxx	<ul style="list-style-type: none"> <li>• Defines the IMSID tracked by the active FDBR.</li> <li>• Defines whether FDBR applies to DEDB areas defined as SHARELVL=0 1.</li> <li>• Defines XCF group name for the active IMS system and the FDBR tracking region.</li> <li>• Specifies the number of seconds before FDBR goes into time-out status.</li> <li>• Specifies whether control blocks for DEDB processing are allocated in private storage or ECSA.</li> </ul>

Table 36. Tailoring Actions for PROCLIB Members for Batch Executions (continued)

PROCLIB Member	Tailoring Action
DLISAS	<ul style="list-style-type: none"> <li>• Adds full-function database DD statements and removes the corresponding statements from the IMS procedure. (No changes are necessary for dynamically allocated databases).</li> <li>• Specifies sizes of two PSB pools using the DLIPSB and CSAPB parameters of the IMS procedure.</li> <li>• Moves the pools to SYS1.PROCLIB.</li> <li>• If the procedure is renamed, specifies this name with the DLINM parameter of the IMS procedure.</li> </ul>
IMSMMSG	<ul style="list-style-type: none"> <li>• Adjusts JOB statement parameters for DFSMPR execution.</li> <li>• Moves to IMS.JOBS</li> </ul>
DFSMPR	<ul style="list-style-type: none"> <li>• Checks and specifies EXEC PARM parameters.</li> </ul>
IMSBATCH	<ul style="list-style-type: none"> <li>• Checks and specifies EXEC PARM parameters.</li> <li>• Add z/OS DD statements.</li> </ul>
DFSVSMxx	<ul style="list-style-type: none"> <li>• Defines buffer pools and other options.</li> <li>• Allows use of sequential buffering.</li> <li>• Specifies initial WADS and OLDS and dual logging.</li> <li>• Coordinates with IMS procedure.</li> <li>• Enables long busy handling.</li> </ul>
DFSHSBxx	<ul style="list-style-type: none"> <li>• Establishes IMS active and alternate subsystems in XRF complex.</li> </ul>
DFSDRFxx	<ul style="list-style-type: none"> <li>• Defines DREF requirements list.</li> </ul>
DFSFIXxx	<ul style="list-style-type: none"> <li>• Defines page fix list.</li> </ul>
DFSINTxx	<ul style="list-style-type: none"> <li>• Defines, loads, and executes user-written exits during dependent region initialization.</li> </ul>
DFSMPLxx	<ul style="list-style-type: none"> <li>• Defines preload program list.</li> </ul>
DFSSQxxx	<ul style="list-style-type: none"> <li>• Defines PROCLIB member for CQS address space.</li> <li>• Defines subsystem name for CQS address space.</li> <li>• Defines name for shared expedited message handler queues.</li> <li>• Defines primary structure containing shared message queues.</li> <li>• Defines XCF IMS shared queues group name.</li> </ul>
IMSWTnnn	<p>Adjusts JOB statement parameters for DFSWTnnn execution. Moves to IMS.JOBS.</p>
DFSWTnnn	<ul style="list-style-type: none"> <li>• Checks and specifies SYSOUT class for spooled output.</li> </ul>
DXRJPROC	<ul style="list-style-type: none"> <li>• Establishes procedures for each required IRLM address space</li> </ul>
DFSPBIMS	<ul style="list-style-type: none"> <li>• Renames to DFSPBxxx, where xxx is determined by IMS procedure RGSUF=xxx.</li> <li>• Defines IMS control region execution parameters.</li> </ul>
DFSPBDBC	<ul style="list-style-type: none"> <li>• Renames to DFSPBxxx, where xxx is determined by DBC procedure RGSUF=xxx.</li> <li>• Defines DBC control region execution parameters.</li> </ul>

Table 36. Tailoring Actions for PROCLIB Members for Batch Executions (continued)

PROCLIB Member	Tailoring Action
DFSPBDCC	<ul style="list-style-type: none"> <li>• Renames to DFSPBxxx, where xxx is determined by DCC procedure RGSUF=xxx</li> <li>• Defines DCC control region execution parameters.</li> </ul>

## Controlling Procedure Library Modifications

Using commands to name a procedure keeps the initial operating instructions simple and avoids complex symbolic parameter data entry; however, you must control the content of the JCL residing in IMS.PROCLIB in this case. For example, if the MTO enters:

```
/START REGION IMSBCH1
```

the procedure IMSBCH1 must be correctly coordinated to a known BMP, to an appropriate PSB and transaction queue and to other system options and identifying parameters.

Your control responsibilities might include auxiliary procedures for database reorganization, recovery, or system output control; furthermore, you must coordinate the modification level of each procedure to the actual production environment. For example, if an application program is modified and requires more dependent region storage, you must coordinate the program library and IMS.PROCLIB changes. A suggested technique is to include JCL comment statements that document the date and the reason for the change.

Be careful in making physical changes in procedure library members. Many of the DD statements extend over several input records and involve positional parameters. You can use a data dictionary to record and maintain the procedure library members. You can then check the changes at the terminal or review printed listings of the changed members.

## IMS Syntax Checker

Use the IMS Syntax Checker to avoid typographical and syntactical errors when modifying the parameter values for the DFSPBxxx, DFSDCxxx, and DFSSQxxx IMS.PROCLIB members. The Syntax Checker ensures that the parameters you use are valid prior to shutting down and re-starting your IMS control regions. This validity check helps avoid restart errors.

The Syntax Checker:

- Reads in the PROCLIB member
- Displays the parameters and values
- Allows modification of the parameter values
- Helps migrate supported PROCLIB members from one version of IMS to another
- Displays default values for parameters
- Verifies the validity of parameters and values
- Saves changes back to the PROCLIB member

**Requirement:** The IMS control region must be restarted to activate the new parameter values.

Appendix C, “Syntax Checker Example,” on page 529 contains more information about the Syntax Checker, including sample panels.

### Preparing for IMS Job Execution

Before executing the IMS procedure as a system task:

1. Move the following procedures to a system procedure library:
  - IMS
  - IMSRDR
  - DBC
  - DBRC
  - DLISAS
  - CQS
2. Enable message region and spool output jobs:
  - Tailor IMSMSG for message region to your requirements
  - Tailor IMSWTnnn for spool output to your requirements
3. Move IMSMSG and IMSWTnnn to IMS.JOBS
4. Start the message regions and spool output jobs using one of the following:
  - IMS commands
  - System commands from the system console

---

## Specifying IMS PROCLIB Members

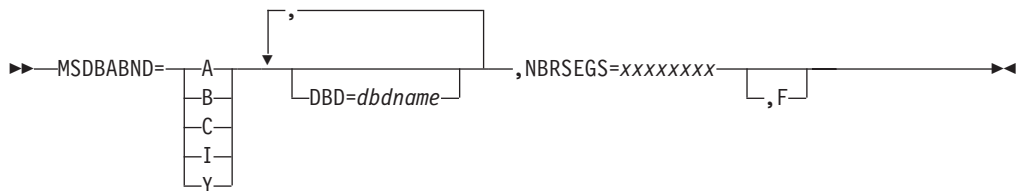
This section describes the IMS.PROCLIB members and provides reference information about control statement syntax and parameter descriptions.

### Member DBFMSDBx

This member is not applicable in the DBCTL environment.

This procedure allows you to specify the Main Storage Databases (MSDBs) to be loaded at startup. You must have previously placed this procedure in the library whose DD name is PROCLIB. Several DBFMSDBx procedures can be present in the library; differentiate them by varying the last character (the suffix, x) of the name. The MSDB= keyword parameter of the IMS procedure specifies the suffix of the particular procedure to be used at startup.

A DBFMSDBx procedure consists of one or more 80-character records, each specifying an MSDB to be loaded. The format is:



#### MSDBABND=

Specifies that the IMS control region is to abend if an error occurs during MSDB loading at system initialization. The valid values for the MSDBABND= parameter are:

- A** Valid for both I and C options.
- B** Valid for both Y and C options.
- C** Abend if initial checkpoint for MSDBs cannot occur.

- I** Abend for one or more of the following reasons:
- No segments exist in the MSDBINIT data set for at least one defined MSDB.
  - For the reasons described in option Y.
- Y** Abend if the MSDBs cannot be loaded due to errors with the MSDBINIT data set.

**DBD=**

Specifies the same one- to eight-character DBD name as was specified at DBDGEN. Multiple DBD= parameters are allowed.

**NBRSEGS=**

Specifies the number of database records expected for this MSDB. You must specify a one- to eight-digit number greater than or equal to the number of MSDB segments loaded at restart. By specifying a number greater than the number of segments to be loaded, the parameter can be used to reserve space for a terminal-related dynamic MSDB.

- F** Optionally specifies that the MSDB is to be page-fixed.

You can separate keyword parameters with either commas or blanks. Positions 73 through 80 are ignored but can be used to sequence the records. Syntax errors are identified by messages.

At startup, the MSDBs specified by a DBFMSDBx procedure are loaded from one of the sequential data sets identified as MSDBINIT, MSDBCP1, or MSDBCP2. You cannot, therefore, use the DBFMSDBx procedure to specify MSDBs that are not included in these data sets; an abend occurs if you use DFMSDBx in this manner.

**Member DFSCGxxx**

The DFSCGxxx PROCLIB member specifies parameters related to the Common Service Layer (CSL), including the Operations Manager (OM), the Resource Manager (RM), and the Structured Call Interface (SCI). The suffix is specified on the CSLG= parameter. All IMSplex members (OM, RM, SCI, IMS, and others) that are in the same IMSplex sharing group, sharing either data bases or message queues must specify the same values.

A DFSCGxxx member consists of one or more 80-character records. Keyword parameters are coded in positions 1-71 in free format and may contain leading and trailing blanks. Positions 72-80 are ignored. Multiple keywords may be specified in each record; commas are used to delimit keywords.

For installations that do not require Resource Manager services, the DFSCGxxx PROCLIB member includes parameters (RMENV, OMPROC, and SCIPROC) for specifying that a Resource Manager environment is not required. These parameters are useful for customers who do not use shared queues or a sysplex, but would like to use enhanced command processing options that are available with IMS type-1 commands.

Table 37 identifies the DFSCGxxx parameters and their applicability based on whether or not RM is used.

*Table 37. Applicability of DFSCGxxx Parameters Based on Resource Manager Environment*

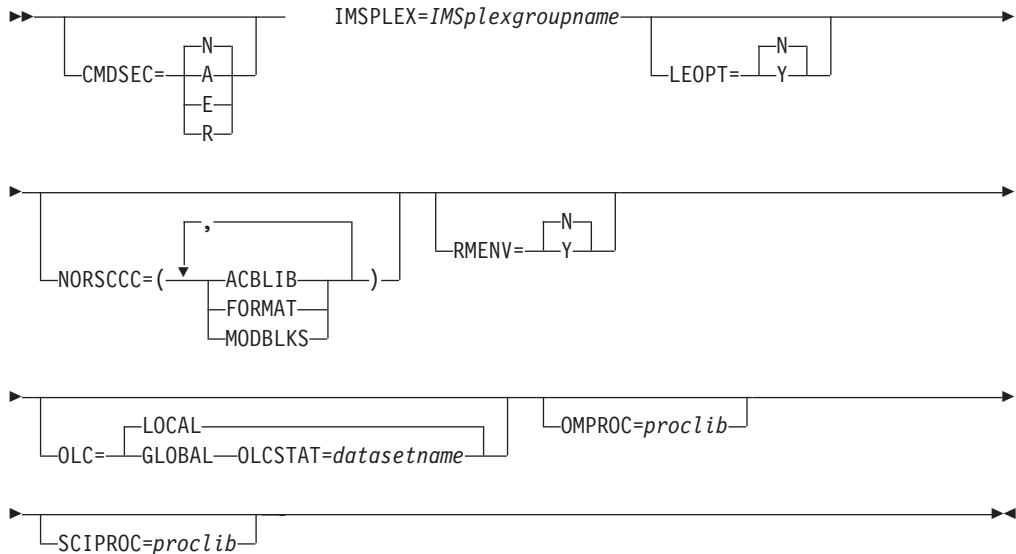
Parameter	RMENV=Y	RMENV=N
CMDSEC	optional	optional



Table 37. Applicability of DFSCGxxx Parameters Based on Resource Manager Environment (continued)

Parameter	RMENV=Y	RMENV=N
IMSPLEX	required	required
LEOPT	optional	optional
NORSCCC	optional	not applicable
OLC	optional	optional
OLCSTAT	optional	optional
OMPROC	not applicable	optional
RMENV	optional	required
SCIPROC	not applicable	optional

**Important:** When RMENV=N and OLC=GLOBAL is specified, ensure that no other IMS attempts to use the same OLCSTAT data set. If another IMS uses the same OLCSTAT Data Set, or if the OLCSTAT data set utility (DFSUOLC0) is used to initialize the data set for another member, an additional IMSID is added to the OLCSTAT data set. If this occurs and IMS (running with RMENV=N and OLC=GLOBAL specified) attempts to initiate an online change by issuing the INIT | TERM OLC command, the command is rejected because an IMSID other than the one executing the command resides in the OLCSTAT data set. The OLCSTAT data set must then be corrected before an INIT | TERM OLC can complete successfully.



**CMDSEC=**

Specifies whether or not IMS should perform security checking on commands that are routed from OM, and if so, whether RACF or the IMS Command Security Exit (or both) should be used. If IMS performs the security checking on commands, the CMDSEC keyword also specifies whether RACF, the IMS Command Authorization Exit (DFSCCMD0), or both should be used. This option only applies to the commands that can be issued through the OM API and are in the original “classic” IMS command syntax format (allowed from the IMS master terminal).

For commands that are in the enhanced IMSplex command syntax (that are only allowed through the OM API, not allowed from the IMS master terminal), all security checking is performed by OM and is controlled by the CMDSEC= parameter on the OM startup JCL or on the OM initialization PROCLIB member (CSLOIxxx).

**Related Reading:** For all the details of the CSLOIxxx PROCLIB member, see the *IMS Version 9: Common Service Layer Guide and Reference*.

**Recommendation:** IBM recommends that OM command security be used instead of IMS security. By allowing OM to perform the security checks, commands which fail security authorization are not routed to IMS, which reduces processing overhead and network traffic. When IMS command security is used, it is up to the user to ensure that all IMS systems use the same security profiles and/or user exits. If IMS systems in the same IMSplex use different security rules, the results of command security checking may be unpredictable.

**A** Specifies that both RACF and DFSCCMD0 are to be called (options E and R). RACF is called first. Then the security authorization facility (SAF) return code, RACF return code, and RACF reason code are passed to DFSCCMD0.

**Related Reading:** For complete information about the Command Authorization Exit (DFSCCMD0), see the *IMS Version 9: Customization Guide*.

**E** Specifies that the IMS command security exit, DFSCCMD0, is to be called for command authorization.

**N** Specifies that no authorization checking is to be done. Security checking can still be performed by OM depending on the specification of the OM CMDSEC= parameter.

**R** Specifies that RACF is to be called for command authorization.

#### **IMSPLEX=**

Specifies a one- to five-character identifier. IMS appends this identifier to “CSL” to create the IMSplex group name. All IMSplex members (OM, RM, IMS, CQS, etc) that are in the same IMSplex sharing group sharing either data bases or message queues must specify the same identifier. IMS has no way to enforce this; the user is responsible for ensuring that the IMSplex name is correctly specified. The same identifier must also be used for the IMSPLEX= parameter in the CSLSIxxx, CSLOIxxx, CSLRIxxx, and CQSIPxxx PROCLIB members.

When RMENV=N, IMSPLEX= specifies a one- to five-character name of the IMSplex that OM, SCI, and the IMS control regions join. This IMSplex is simply a grouping of all address spaces that are managed by one or more OMs.

This parameter is required.

#### **LEOPT=Y | N**

Specifies whether or not IMS should allow Language Environment (LE) dynamic runtime parameter overrides.

#### **NORSCC=()**

Specifies that no resource consistency checking is to be performed for the specified resources. The resources to not check for consistency must be enclosed within parentheses and separated by commas. The NORSCC keyword must precede the left parenthesis. NORSCC is an optional parameter.

If a resource structure is defined for the IMSplex, by default, consistency checking is performed for ACBLIB, FORMAT, and MODBLKS. Consistency checking is performed on the data set names; it is not performed on the resources that reside in the libraries. Resource consistency checking is useful for cloned systems that either share all of these data sets or use the same data set names on each IMS.

If no resource structure is defined, no resource definition consistency checking is performed.

When RMENV=N, the NORSCCC parameter is ignored.

Specify one or more of the following parameters:

#### **ACBLIB**

The ACBLIB data set names are not checked for consistency. The IMSs in the IMSplex do not need to define the same data sets for the ACBLIB library. ACBLIB only applies if OLC=GLOBAL is specified.

#### **FORMAT**

The FORMAT data set names are not checked for consistency. The IMSs in the IMSplex do not need to define the same data sets for the FORMAT library. FORMAT only applies if OLC=GLOBAL is specified.

#### **MODBLKS**

The MODBLKS and MATRIX data set names are not checked for consistency. The IMSs in the IMSplex do not need to define the same data sets for the MODBLKS and MATRIX library. MODBLKS only applies if OLC=GLOBAL is specified.

#### **OLC=LOCAL | GLOBAL**

Specifies the scope of the online change. OLC=LOCAL means the online change applies locally to each IMS. Local online change is prepared and committed by using the /MODIFY PREPARE and /MODIFY COMMIT commands on each local IMS. The local online changes must be manually coordinated across an IMSplex. OLC=GLOBAL means that the online change is coordinated across the IMSplex. Global online change is prepared and committed by using the INITIATE OLC command. If global online change is enabled and a resource structure is defined, the MODBLKS, MATRIX, FORMAT, and ACBLIB data sets must be consistent across the IMSplex, unless resource consistency checking is omitted via the NORSCCC keyword.

OLC=GLOBAL is not supported for RSR tracking IMSs. If specified, IMS abends during restart with ABENDU2801 subcode X'0013'.

#### **OLCSTAT=**

Specifies the 1 to 44 character data set name of the OLCSTAT data set. OLCSTAT is ignored if OLC=LOCAL is specified. OLCSTAT is required if OLC=GLOBAL is defined. The OLCSTAT data set is a catalogued BSAM data set that contains global online change information and status. The data set name is used to dynamically allocate the data set when an IMS initialized, restarts, or is master of an online change phase.

All IMSs in an IMSplex must refer to the same physical OLCSTAT data set. If a resource structure is defined to the IMSplex, IMS insures that the OLCSTAT data set name is defined consistently for the IMSplex. If the OLCSTAT data set name definition is not consistent with the OLCSTAT defined to other IMSs in the IMSplex, IMS initialization fails.

#### **RMENV=Y | N**

Indicates whether (Y) or not (N) IMS requires an RM environment.

- Y** Indicates that IMS requires an RM environment in order to use Resource Manager services. IMS initialization does not complete until IMS successfully registers with RM. This is the default.
- N** Indicates that IMS does not require an RM environment and will not use Resource Manager services. IMS does not attempt to register with RM even if an RM address space is active. When RMENV=N, you do not need to define or start an RM address space. IMS commands and functions that require RM are not available in this environment.

When RMENV=N, online change is as follows:

- If OLC=LOCAL, use the /MODIFY command to initiate online change. The MODSTAT data set must be defined.
- If OLC=GLOBAL, use the INIT OLC command to initiate online change. The OLCSTAT data set must be defined; however, it cannot be shared between IMS systems.

You can change the value of RMENV= across IMS warm or cold starts.

**OMPROC=**

Specifies the one- to eight-character name of the PROCLIB member that contains the procedure for the OM address space. This parameter is only valid when RMENV=N is specified. If RMENV=Y is specified, this parameter is ignored. This parameter is optional.

If OMPROC= is not specified, start the OM address space before starting IMS. This start sequence can be performed through an automation program or by another IMS control region (if RMENV=N is specified).

**SCIPROC=**

Specifies the one- to eight-character name of the PROCLIB member that contains the procedure for the SCI address space. This parameter is only valid when RMENV=N. If RMENV=Y is specified, this parameter is ignored. This parameter is optional.

If SCIPROC= is not specified, start the SCI address space before starting IMS. This start sequence can be performed through an automation program or by another IMS control region (if RMENV=N).

**Member DFSDCxxx**

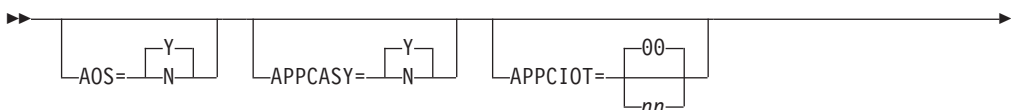
The DFSDCxxx PROCLIB member applies to the DB/DC and DCCTL environments. You can find an example of the DFSDCxxx PROCLIB member, DFSDC000, in library SDFSSLIB.

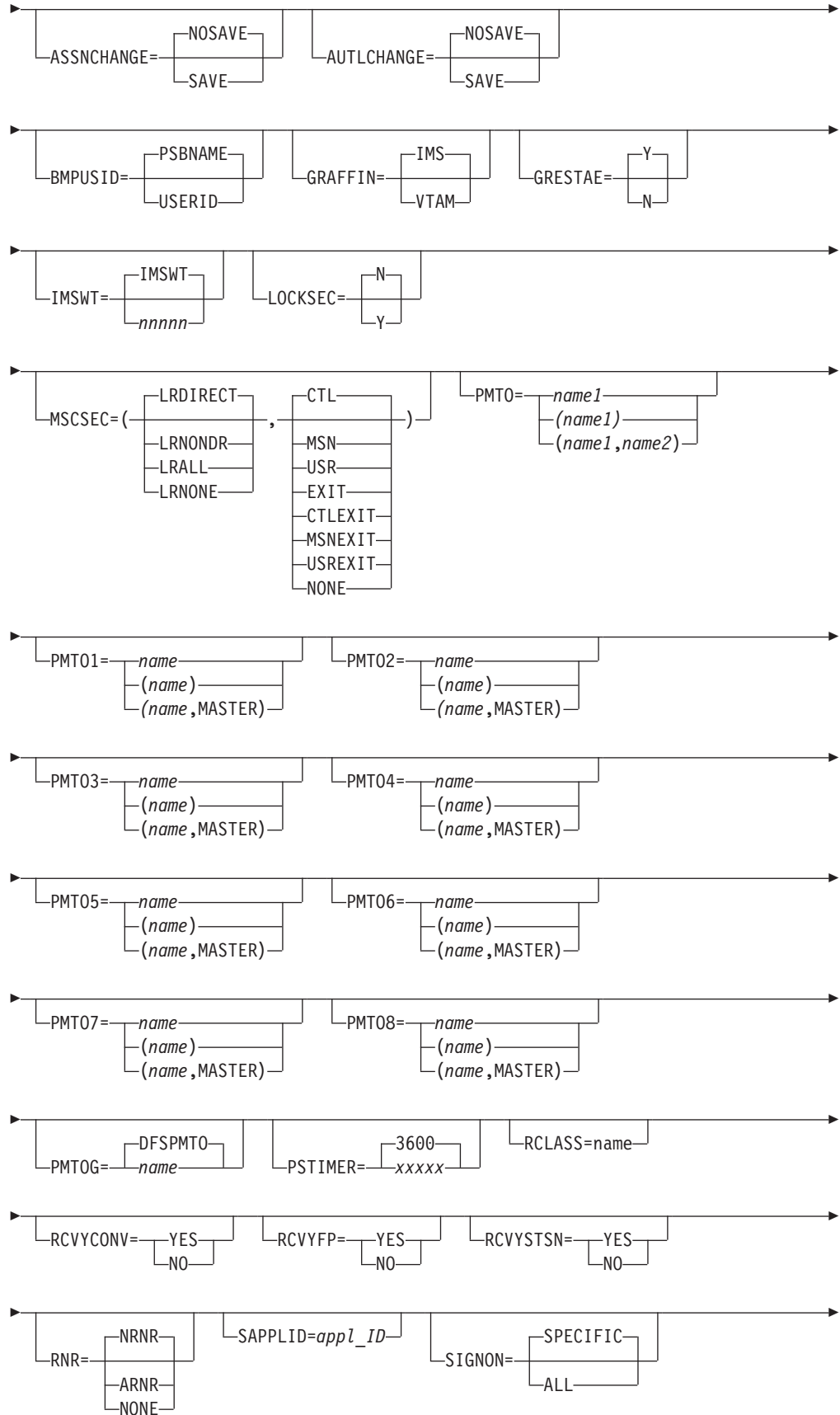
**Specifying the Data Communications Options**

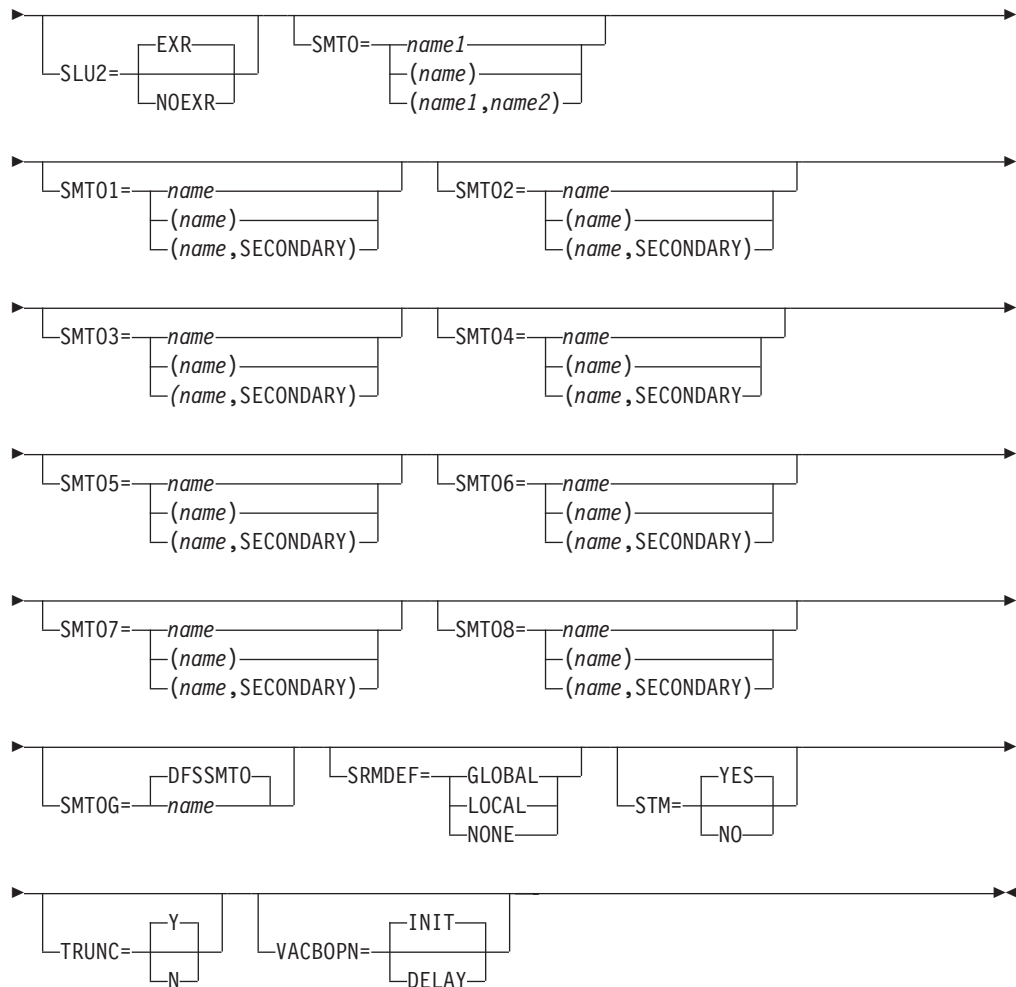
The DC= parameter in the IMS or DCC startup procedure defines the DFSDCxxx PROCLIB member that is used. The default suffix is 000.

**Control Statements**

A DFSDCxxx member consists of one or more 80-character records. You code keyword parameters in positions 1-71. Positions 72-80 are ignored. Keywords can contain leading and trailing blanks. If you specify multiple keywords in a record, use commas as delimiters. The format is:





**AOS=**

Specifies whether (Y) or not (N) shared message queue support for synchronous APPC/OTMA is active. The default is active (Y). Shared message queue support for APPC/OTMA allows IMSplex users to execute transactions that are originated by APPC or OTMA on a back-end system.

**Important:** Specifying N deactivates APPC/OTMA shared message queue support for all of the members within the IMSplex.

**APPCASY=**

Specifies whether a non-response transaction originating from a program-to-program switch should be scheduled asynchronously. The default is APPCASY=Y. IBM recommends that you keep this specification.

**APPCIOT=**

Specifies the APPC time-out value expressed in minutes. Valid values are between 00 and 1440. If APPCIOT=00, time-out detection is inactive.

**Requirement:** Changing this parameter after a previous IMS start does not require a cold start. You must perform a warm start, however, to allow the new specification to take effect.

**ASSNCHANGE=NOSAVE | SAVE**

Sets the default value for the /ASSIGN LTERM|USER TO USER command. SAVE specifies that LTERM assignments (made in ETO) will be retained across session and IMS restarts (the user and LTERM control blocks will not be

deleted). The default is NOSAVE, which indicates that the assignment will not be retained and the user and LTERM control blocks will be deleted.

You can override the ASSNCHANGE control statement by specifying SAVE or NOSAVE on the /ASSIGN command. If you issue the /ASSIGN command without specifying SAVE or NOSAVE, IMS will use the default value that is specified on the ASSNCHANGE control statement.

**Requirement:** Changing this parameter after a previous IMS start does not require a cold start. You must perform a warm start, however, to allow the new specification to take effect.

#### **AUTLCHANGE= NOSAVE | SAVE**

Sets the default value for the /CHANGE USER AUTOLOGON command. SAVE specifies that autologon changes made in ETO will be retained across session and IMS restarts (the user control block will not be deleted and will remain in effect across a restart or XRF takeover until another /CHANGE command with the NOSAVE parameter is issued). The default is NOSAVE, which indicates that the autologon changes will not be retained across session and IMS restarts (the user control block is deleted).

You can override the AUTLCHANGE control statement by specifying SAVE or NOSAVE on the /CHANGE command. If you issue the /CHANGE command without specifying SAVE or NOSAVE, IMS will use the default value that is specified on the ASSNCHANGE control statement.

**Requirement:** Changing this parameter after a previous IMS start does not require a cold start. You must perform a warm start, however, to allow the new specification to take effect.

#### **BMPUSID=PSBNAME | USERID**

Specifies the system option for the value to be placed in the user ID field of the message prefix for a message generated by a non-message driven BMP and the value to be used for the userid for CHNG and AUTH calls made by a non-message-driven BMP.

- If BMPUSID=USERID is specified, the value from the USER= keyword on the JOB statement is used.
- If USER= is not specified on the JOB statement, the program's PSB name is used.
- If BMPUSID=PSBNAME is specified, or if BMPUSID= is not specified at all, the program's PSB name is used.

**Note:** This parameter is not valid for DBCTL. The value from the USER= keyword on the JOB statement is used for a JBP.

**Requirement:** Changing this parameter after a previous IMS start does not require a cold start. You must perform a warm start, however, to enable the new specification to take effect.

#### **GRAFFIN=IMS | VTAM**

Specifies whether IMS or VTAM should manage generic resource affinities.

IMS indicates that IMS manages the generic resource affinities. This is the default.

VTAM indicates that VTAM manages the generic resource affinities. You can use VTAM to manage LU6.1 (ISC) affinity.

Note that if you enter the keyword or parameter for this option incorrectly, IMS uses the default and does not generate an error message.

IMS ignores the GRAFFIN keyword and sets the VTAM affinity for each session during terminal or user logon if the following is true:

- Your operating system is z/OS V1R2 (or higher)
- You are using VTAM Generic Resources (VGR)

**Related Reading:** For details about how IMS sets the VGR affinity, see the *IMS Version 9: Administration Guide: Transaction Manager*.

**Requirement:** Changing this parameter after a previous IMS start does not require a cold start, although a cold start is recommended in order to clear outstanding VTAM Generic Resource terminal affinities. You must perform at least a warm start to allow the new specification to take effect.

#### **GRESTAE=Y | N**

Specifies whether IMS should bypass the VTAM generic resource logic in the IMS ESTAE exit.

Y indicates that IMS should continue the existing ESTAE logic to delete affinity for all nodes where no status remains before closing the ACF/VTAM ACB.

N indicates that IMS should close the ACF/VTAM ACB immediately to expedite IMS termination, and leave affinity for all nodes set.

**Requirement:** Changing this parameter after a previous IMS start does not require a cold start. You must perform a warm start, however, to allow the new specification to take effect.

#### **IMSWT=IMSWT | nnnnn**

Specifies up to 5 characters that will be appended with 3-character line number yyy to make up the full JOBNAME (nnnnnyyy) in the /STA REGION jobname for the auto-scheduled spool print utility. If IMSWT= is not specified or if nnnnn is blank, IMS uses the default value of IMSWT (IMSWTyyy).

**Requirement:** Changing this parameter after a previous IMS start requires a cold start to take effect.

#### **LOCKSEC=Y | N**

When password protection is used for the /LOCK and /UNLOCK commands, specifies whether (Y) or not (N) calls to RACF security and the Transaction Authorization exit routine (DFSCTRNO) are made after optional SMU calls. When LOCKSEC=Y is specified, the command is processed if either of the following is true:

- The resource specified by the command is defined to RACF and the user is authorized to use the resource
- The resource is not defined to RACF

If the resource is defined to RACF, but the user is not authorized to use the resource, the command is rejected with the message DFS3689W.

#### **MSCSEC=**

Specifies the security options for direct and non-direct routed transactions received from an MSC link.

##### **LRDIRECT**

Specifies that Link Receive routing is to use RACF to check security for direct routed transactions. SMU security is used to check security for non-direct routed transactions. This is the default.

##### **LRNONDR**

Specifies that Link Receive routing is to use RACF to check security for



non-direct routed transactions. No security checking is performed for direct routed transactions. The IMS subsystem does not call the SMU for either type of transaction.

**LRALL**

Specifies that Link Receive routing is to use RACF to check security for both direct and non-direct routed transactions. The IMS subsystem does not call SMU for either type of transaction.

**LRNONE**

Specifies that the IMS subsystem does not request any security checking for either type of transaction.

**CTL**

Specifies that RACF security is to authorize the use of the transaction based upon the user ID of the IMS control region.

**MSN**

Specifies that RACF security is to authorize the use of the transaction based upon the user ID of the MSNAME.

**USR**

Specifies that RACF security is to authorize the use of the transaction based upon the user ID of the inputting terminal.

**EXIT**

Specifies that authorization is performed by the Transaction Authorization exit routine (DFSCTRN0).

**CTLEXIT**

Specifies that both RACF security and the Transaction Authorization exit routine (DFSCTRN0) are used to authorize the IMS control region user ID to use the transaction.

**MSNEXIT**

Specifies that both RACF security and the Transaction Authorization exit routine (DFSCTRN0) are used to authorize the MSNAME to use the transaction.

**USREXIT**

Specifies that both RACF security and the Transaction Authorization exit routine (DFSCTRN0) are used to authorize the user ID of the inputting terminal to use the transaction.

**NONE**

Specifies that no security authorization checking is performed.

**PMTO=*name1* | *name2***

Specifies the node name of the primary master terminal (PMTO). This name overrides the PMTO node name specified during IMS system definition. If you are using a BTAM PMTO, and specify this parameter, it is ignored.

**Requirement:** Changing this parameter after a previous IMS start requires a cold start to take effect.

**Restrictions:** See "**Restrictions**" on page 349.

**PMTO1-8=*name1* | *name1*,MASTER**

Specifies the first through the eighth LTERM names of the PMTO. The names you specify here override the first through the eighth LTERM names specified during IMS system definition. The LTERM name specified with the MASTER designation becomes the PMTO LTERM. These LTERM names must be unique for each IMS system within a shared IMS environment.

**Requirement:** Changing this parameter after a previous IMS start requires a cold start to take effect.

**Restrictions:** See “**Restrictions**” on page 349.

#### **PMTOG=DFSPMTO** *Iname*

Specifies the generic LTERM name of the PMTO. The default is DFSPMTO. Use the same generic LTERM name for each IMS system within a shared IMS environment.

**Requirement:** Changing this parameter after a previous IMS start requires a cold start to take effect.

**Restrictions:** See “**Restrictions**” on page 349.

#### **PSTIMER=3600** | *xxxxx*

When Rapid Network Reconnect (RNR) is used, the PSTIMER= parameter specifies a one- to five-digit number indicating the number of seconds IMS can wait pending recovery for persistent sessions following an IMS or ACF/VTAM failure. The restarted IMS must successfully issue an OPEN ACB before the timer expires or the sessions will be terminated. Valid values are from zero (no timer) to 86400 seconds (24 hours). The IMS default is 3600 seconds (1 hour). If a null or invalid value is specified, IMS uses 3600.

When an ACB name is specified on the MNPS= keyword (in the PROCLIB member DFHSBxx or DFSPBxxx, or in the IMS procedure), specifies the amount of time before the XRF alternate must successfully open the MNPS ACB. This applies to an XRF takeover only. It does not apply to an emergency restart because persistent sessions are not restored following a restart in an XRF system using MNPS.

**Requirement:** Changing this parameter after a previous IMS start does not require a cold start. You must perform a warm start, however, to allow the new specification to take effect.

#### **RCLASS=name**

Specifies an identifier of one to seven alphanumeric characters that is to be used to identify the IMS system as a resource class to RACF for transaction authorization and user ID verification. “IMS” is the default.

The specification is valid only if one of the following is true:

- TYPE=RACFTERM is specified on the SYSGEN SECURITY macro; or RCF=A,T, or Y is specified on the IMS procedure.
- TYPE=RACFAGN, RASRACF, or RAS is specified on the SYSGEN SECURITY macro, or ISIS=1, R, or A is specified on the IMS procedure.

IMS class names do not need to be unique. If you do make them unique, you can, for example, define the same transaction name on a production subsystem and a test subsystem, with different access lists for each subsystem, with no ambiguity.

The RCLASS= specification on the DFSDCxxx PROCLIB member overrides the RCLASS= specification on the SYSGEN SECURITY macro.

#### **RCVYCONV=**

Specifies whether the status of a conversation can be recovered (YES) or not (NO). RCVYCONV applies to conversation status, not to output messages. Even if the conversation status is not recovered, conversation output continues to be recoverable and is delivered asynchronously.

Table 38 describes the default values for RCVYCONV as they relate to the values specified on the SRMDEF keyword.

Table 38. RCVYCONV Values Related to SRMDEF Values

	RCVYCONV=YES	RCVYCONV=NO
SRMDEF=GLOBAL	Valid (default)	Valid
SRMDEF=LOCAL	Valid (default)	Valid
SRMDEF=NONE	Invalid	Valid (default)

If RCVYCONV is specified incorrectly, message DFS1920I is issued and IMS uses the appropriate default from Table 38.

#### RCVYFP=

Specifies whether the status of Fast Path can be recovered (YES) or not (NO). RCVYFP applies to conversation status and Fast Path output.

Table 39 describes the default values for RCVYFP as they relate to the values specified on the SRMDEF keyword.

Table 39. RCVYFP Values Related to SRMDEF Values

	RCVYFP=YES	RCVYFP=NO
SRMDEF=GLOBAL	Valid (default)	Valid
SRMDEF=LOCAL	Valid (default)	Valid
SRMDEF=NONE	Invalid	Valid (default)

**Restrictions:** If SRMDEF=LOCAL, STSN recoverability (RCVYSTSN) and fast path recoverability (RCVYFP) must be the same. They must both specify YES or both specify NO.

If RCVYFP is specified incorrectly, message DFS1920I is issued and IMS will use the appropriate default from Table 39.

**Related Reading:** For a discussion on the recoverability of Fast Path status, see the section titled “Fast Path Recovery” in the “Resource Modes and States” section in the *IMS Version 9: Administration Guide: Transaction Manager*.

If RCVYFP is specified incorrectly, message DFS1920I is issued and IMS will use the appropriate default from Table 39.

#### RCVYSTSN=

Specifies whether the status of STSN terminals (SLUP, FINANCE, and ISC) can be recovered (YES) or not (NO).

Table 40 describes the default values for RCVYSTSN as they relate to the values specified on the SRMDEF keyword.

Table 40. RCVYSTSN Values Related to SRMDEF Values

	RCVYSTSN=YES	RCVYSTSN=NO
SRMDEF=GLOBAL	Valid (default)	Valid
SRMDEF=LOCAL	Valid (default)	Valid
SRMDEF=NONE	Invalid	Valid (default)

**Restrictions:** If SRMDEF=LOCAL, STSN recoverability (RCVYSTSN) and fast path recoverability (RCVYFP) must be the same. They must both specify YES

or both specify NO. If SRMDEF=GLOBAL, Fast Path is always non-recoverable, so if there is a failure during the execution of a fast path transaction, the STSN session must be restarted cold.

If RCVYSTSN is specified incorrectly, message DFS1920I is issued and IMS will use the appropriate default from Table 40 on page 345.

#### **RNR=NRNR | ARNR | NONE**

Specifies that IMS should enable Rapid Network Reconnect (RNR) support for ACF/VTAM persistent sessions as defined on the ACF/VTAM APPL statement for IMS. This parameter is used as a system default for an unspecified RNR value for the OPTIONS= keyword at the terminal level.

**ARNR** Specifies that a basic session reconnect should be done automatically by IMS following an IMS/ESA or ACF/VTAM restart. This level of session reconnect is to a data traffic reset point equivalent to logon without available signon data. A DFS3649 signon required message can be sent as appropriate for the terminal type and definition.

**NRNR** Specifies that automatic session reconnect should not be done. The session will be terminated following restart of IMS or ACF/VTAM.

**NONE** Specifies that the RNR function should not be activated. This option can also be used to deactivate the RNR function if it was activated prior to the IMS restart.

**Requirement:** Changing this parameter after a previous IMS start does not require a cold start, although a cold start is recommended in order to clear persistent session data in VTAM and terminate IMS sessions that are waiting for an RNR reconnection. You must perform at least a warm start to allow the new specification to take effect.

#### **SAPPLID=appl\_ID**

Specifies the application ID to be used when calling RACF, or equivalent product, during signon. If SAPPLID= is not specified, the IMSID of that IMS is the default application ID. The application ID becomes important when Passtickets are used instead of passwords because the application ID is used in the RACF creation and verification of the Passticket.

**Requirement:** Changing this parameter after a previous IMS start does not require a cold start. You must perform a warm start, however, to allow the new specification to take effect.

#### **SIGNON=SPECIFIC | ALL**

Specifies that either all (ALL) or specific (SPECIFIC) static terminals are required to sign on.

##### **ALL**

Specifies that all static terminals (except the master terminal) are required to sign on. The following devices are not required to signon when SIGNON=ALL is specified:

- LU6.1 devices
- 3284/3286 devices
- SLU1 printer-only devices
- Master terminals

To require master terminals to sign on, specify the OPTIONS=SIGNON on the TYPE or TERMINAL macro.

##### **SPECIFIC**

Specifies that individual static terminals might be required to sign on. These

I terminals will be specified in either system definition with the  
 I OPTIONS=SIGNON parameter, or by SMU definitions with the )(SIGN  
 I statements. SPECIFIC is the default.

**SLU2=EXR | NOEXR**

Specifies whether all IMS-defined SLU2 terminals should be treated as with or without DFT architecture.

**EXR** Specifies that all IMS-defined SLU2 terminals will be treated as with DFT architecture. EXR is the default.

**NOEXR**

Suppresses the SNA exception response that proceeds an IMS error message. As a result of no exception response, the terminal PROG check is prevented, the keyboard is locked, and IMS is allowed to send the error message.

**Requirement:** Changing this parameter after a previous IMS start does not require a cold start. You must perform a warm start, however, to allow the new specification to take effect.

**SMTO=name1 | name2**

Specifies the node name of the secondary master terminal (SMTO). This name overrides the SMTO node name specified during IMS system definition. If you are using a BTAM SMTO, and specify this parameter, it is ignored.

**Requirement:** Changing this parameter after a previous IMS start requires a cold start to take effect.

**Restrictions:** See “**Restrictions**” on page 349.

**SMT01-8=name1 | name1,SECONDARY**

Specifies the first through the eighth LTERM names of the SMTO. The names you specify here override the first through the eighth LTERM names specified during IMS system definition. The LTERM name specified with the SECONDARY designation becomes the SMTO LTERM. These LTERM names must be unique for each IMS system within a shared IMS environment.

**Requirement:** Changing this parameter after a previous IMS start requires a cold start to take effect.

**Restrictions:** See “**Restrictions**” on page 349.

**SMT0G=DFSSMTO | name**

Specifies the generic LTERM name for the secondary master terminal. The default generic SMTO LTERM name is DFSSMTO. Use the same generic LTERM name for all IMS systems within a shared IMS environment.

**Requirement:** Changing this parameter after a previous IMS start requires a cold start to take effect.

**Restrictions:** See “**Restrictions**” on page 349.

**SRMDEF=**

Specifies the status recovery mode for user and node resources. Valid values are GLOBAL, LOCAL, and NONE.

**Related Reading:** In order to understand completely what the GLOBAL, LOCAL, and NONE values mean, you have to first understand how IMS classifies resource status. For a discussion of these concepts, see the *IMS Version 9: Administration Guide: Transaction Manager*.

The following list describes the valid values for SRMDEF.

**GLOBAL**

The significant status of a resource is saved globally in the coupling facility resource structure every time the significant status changes, along with all other recoverable status for that resource. The resource status is restored at the next logon or signon and is available from any IMS system in the IMSplex. Resource status is copied to the local system when that resource becomes active, but is deleted from the local system when it becomes inactive. Status is not recovered in local IMS log records.

**Requirement:** Using a status recovery mode of GLOBAL requires a Resource Manager and a resource structure in the coupling facility. If STM=NO is specified, SRMDEF=GLOBAL is invalid.

**LOCAL**

The significant status is saved in local control blocks and log records, along with all other recoverable resource status. The resource status is restored at the next logon or signon provided that the user or node returns to the same local IMS system. This status is not available on any other IMS system in the IMSplex.

LOCAL mode enforces an affinity for the user or node to the IMS system where the local status exists. This is referred to as RM affinity. IMS will prevent the user or node from accessing any other IMS while local status exists. If the IMS system that has local status fails, the node and user will be allowed access to another IMS system. In this case, the node or user does not recover any status and when the failed IMS restarts, the local status that existed is deleted.

Using a status recovery mode of LOCAL does not require a Resource Manager. If RM is not used to manage node or user resources, then no RM affinity is enforced.

**NONE** The significant status is not saved by RM or in local log records. When the user signs on or the node logs on, the corresponding significant status does not exist.

**STM=YES | NO**

Specifies whether (YES) or not (NO) IMS is to use the RM resource structure to manage TM resources. The resources that are managed include APPC descriptors, VTAM LTERMS, MSNAMEs, VTAM terminal nodes, users, and user IDs. The default is YES.

A cold start is required if STM=NO is specified for installations that were previously using the resource structure.

Any change to the STM= specification requires a cold start of the TM component (COLDCOMM). This restriction is enforced during restart processing through the U233 ABENDU0233.

Note that transactions (both statically defined and CPI-C) are maintained in the resource structure independent of TM resources, regardless of the STM= specification.

If STM= is specified, but IMS is not using a resource structure, the specification is ignored.

**TRUNC=Y | N**

Specifies the system default for the truncated data options used for IMS

conversational transactions. If a transaction definition (TRANSACT) does not specify a truncated data option, the system default is used when a conversation starts for the first time.

**Requirement:** Changing this parameter after a previous IMS start does not require a cold start. You must perform a warm start, however, to allow the new specification to take effect.

**Related Reading:** For more details about truncated data, see *IMS Version 9: Customization Guide*.

### **VACBOPN=INIT | DELAY**

Specifies that the opening of the IMS VTAM ACB is delayed until a /START DC command is issued. This parameter is optional. The following is an example of using the VACBOPN parameter.

An IMS system is restarted (after an emergency restart) and this system includes many Automated Teller Machines (ATMs). All of the ATMs attempt to connect immediately. The logon attempts are rejected by VTAM until IMS opens the VTAM ACB.

If, however, the VTAM ACB is opened during initialization, VTAM begins queuing the logon requests. The programs in the ATMs will wait for a short period of time for a response and then time out. When the timeout occurs, an UNBIND request is sent and this is also queued. This scenario continues until the logon is successful. Meanwhile, IMS has to complete its initialization and then process all of the log records for the emergency restart.

After the successful completion of the emergency restart, the operator can issue a /START DC command to tell VTAM to start accepting logon requests. When this finally happens, there are many UNBIND requests on the queue. At this point, the logon requests are queued behind enough error processing that they also time out.

In some situations, it appears that the IMS system is looping (because no sessions can get connected) and the operator is tempted to terminate IMS. Using the VACBOPN=DELAY parameter will alleviate this problem by delaying the queuing of the logon requests until IMS is ready to start accepting the logon requests (when the /START DC command is issued).

**INIT** Specifies that the IMS VTAM ACB is opened when IMS initializes.

### **DELAY**

Specifies whether the IMS VTAM ACB is opened when a /START DC command is issued.

**Requirement:** Changing this parameter after a previous IMS start does not require a cold start. You must perform a warm start, however, to allow the new specification to take effect.

### **Restrictions:**

- If you use the DFSDCxxx member to override a default name (from system definition), do not specify a statically defined lterm or node name.
- If you use the DFSDCxxx member to override the master or secondary master node names, do not specify a statically defined node name.
- If you use the DFSDCxxx member to override the master or secondary master LTERM names, do not specify a statically defined LTERM name.
- If you use the DFSDCxxx member to override the generic LTERM name of the master LTERM or secondary master LTERM, do not specify a statically defined LTERM name.

- The node that you specify using the PMTO or SMTO parameters must have the same physical characteristics as the default node that you are overriding. For example, you cannot override a SLU2 device with a non-SNA device.
- If you change the PMTO=, SMTO=, PMOTn=, or SMTON= control statements in the DFSDCxxx member, you must perform a cold start of IMS. You can perform a warm start for changes to any other control statements in the DFSDCxxx member.

## Member DFSDRFnn

This member is not applicable in the DBCTL environment.

### Defining DREF Requirements in the IMS Virtual Control Region

You can tune your system to a particular operating environment by specifying that portions of the control region be placed in DREF storage during initialization. You must place control information in the PROCLIB procedure library (usually named IMS.PROCLIB), as part of member DFSDRFnn.

With this member, VTCBs in DREF storage are never paged out to auxiliary storage; they remain in real or expanded storage. DFSDRFnn requires that the system have sufficient expanded storage available for the VTCBs. If you choose the DREF option for VTCBs during installation, but do not have expanded storage, the VTCBs DREF storage are page fixed.

If the VTCBs have been paged out of real storage when a checkpoint occurs, DFSDRFnn can improve the time to bring them back into real storage.

The *nn* suffix of DFSDRFnn is taken from the IMS procedure FIX=*nn* initialization parameter, the same one used for the page fixing member, DFSFIXnn (see "Member DFSFIXnn" on page 352).

If you specify either DFSDRFnn or DFSFIXnn, define the one you do not choose to use as a null member. For example, if you specify FIX=55, define both members (DFSDRF55 and DFSFIX55), and make the member you do not need a null member containing a blank control statement. Without both defined, IMS issues error message DFS0579, indicating a member could not be found. If DFSFIXnn is defined as a null member, message DFS0757 is issued, reminding you that no fixing requests were made.

If BLOCK=VTCB is requested in DFSFIXnn and in DFSDRFnn, the page-fix request in DFSFIXnn is ignored.

**Control Statements:** The control information is contained in 80-character records. Continuation or sequencing must not be entered. The format is as follows:

▶▶—BLOCKS—=—VTCB—▶▶

### BLOCKS=

#### VTCB

Specifies that VTCBs are loaded into DREF storage, but are never to page to auxiliary storage.



## Member DFSFDRxx

This member is applicable to the FDBR region and the FDBR-capable IMS or DBC regions.

### Specifying the FDBR Options

The DFSFDRxx member of IMS.PROCLIB specifies the FDBR options used by the FDR. Multiple DFSFDRxx members can be present in IMS.PROCLIB. The FDRMBR= parameter in the FDR, IMS, or DBC procedure identifies the member currently in effect.

**Control Statement:** The record format is

**Position**

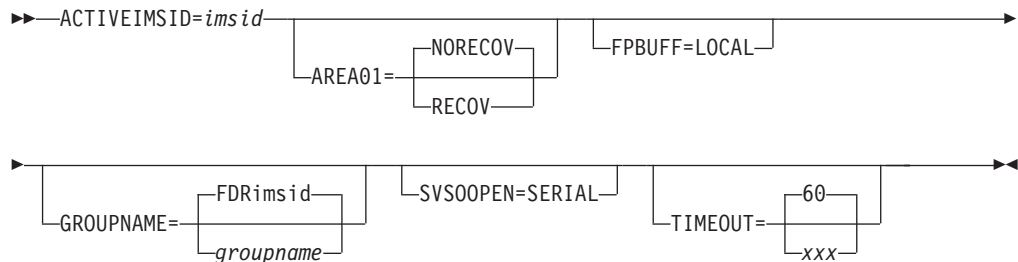
**Contents**

**1-71** Option parameters separated by commas

**72-80** Ignored

The first parameter in a record can have leading blanks. The last parameter in a record is denoted by a comma, followed by one or more blanks. The last parameter in the last record is followed by one or more blanks.

The following parameters can occur in any order in the control statement. Multiple control statements can be provided.



**ACTIVEIMSID=**

Specifies the IMS ID for the subsystem that the active FDBR region tracks. For IMS and DBC procedures (FDBR capable regions) ACTIVEIMSID must match the IMS ID defined in the procedure. This parameter is required.

**AREA01=**

Specifies whether Fast DB Recovery applies to DEDB areas defined as SHARELVL= 0|1. This parameter applies to FDR procedures. IMS and DBC procedures ignore it if it is specified.

**RECOV**

Specifies that Fast DB Recovery applies to SHARELVL= 0|1 areas.

**NORECOV**

Specifies that SHARELVL= 0|1 areas are not recovered by FDBR. Succeeding IMS emergency restarts recover the areas after FDBR completion.

The default is NORECOV.

**FPBUFF=LOCAL**

Specifies that the control blocks for DEDB processing are allocated in the

I FDBR control region private storage. If this option is specified, buffer pools for  
I shared VSO buffers are also allocated in private storage. This keyword is  
I optional.

I If this option is not specified, the control blocks for DEDB processing are  
I allocated in ECSA. This is the default.

#### **GROUPNAME=**

Specifies the XCF group name for the active IMS system and the tracking FDBR region. This parameter is optional. If you do not specify GROUPNAME=, a default name is created by using the ACTIVEIMSID= parameter and the prefix FDR. For example, if the active IMSID is IMS01, the default XCF group name is FDRIMS01.

#### **SVSOOPEN=SERIAL**

I Specifies that all areas requiring redo processing in an FDBR system will be  
I serially processed. This option is ignored for emergency restart and XRF  
I takeover processing. Use this option to reduce the number of structures that are  
I allocated by FDBR for redo processing.

#### **TIMEOUT=**

Specifies the number of seconds that FDBR waits before determining a time-out status for the tracked IMS. Valid values are from 3-999.

The default is 60.

If a value less than 3 is specified, the value 3 is used. If a value greater than 999 is specified, 999 is used.

This parameter is applicable to IMS and DBC procedures.

The value you specify with this parameter applies only to IMS time-out status from XCF. For log surveillance, FDBR uses a five-second time out value, and a three-second delay interval before calling DBRC to check whether an OLDS switch has occurred.

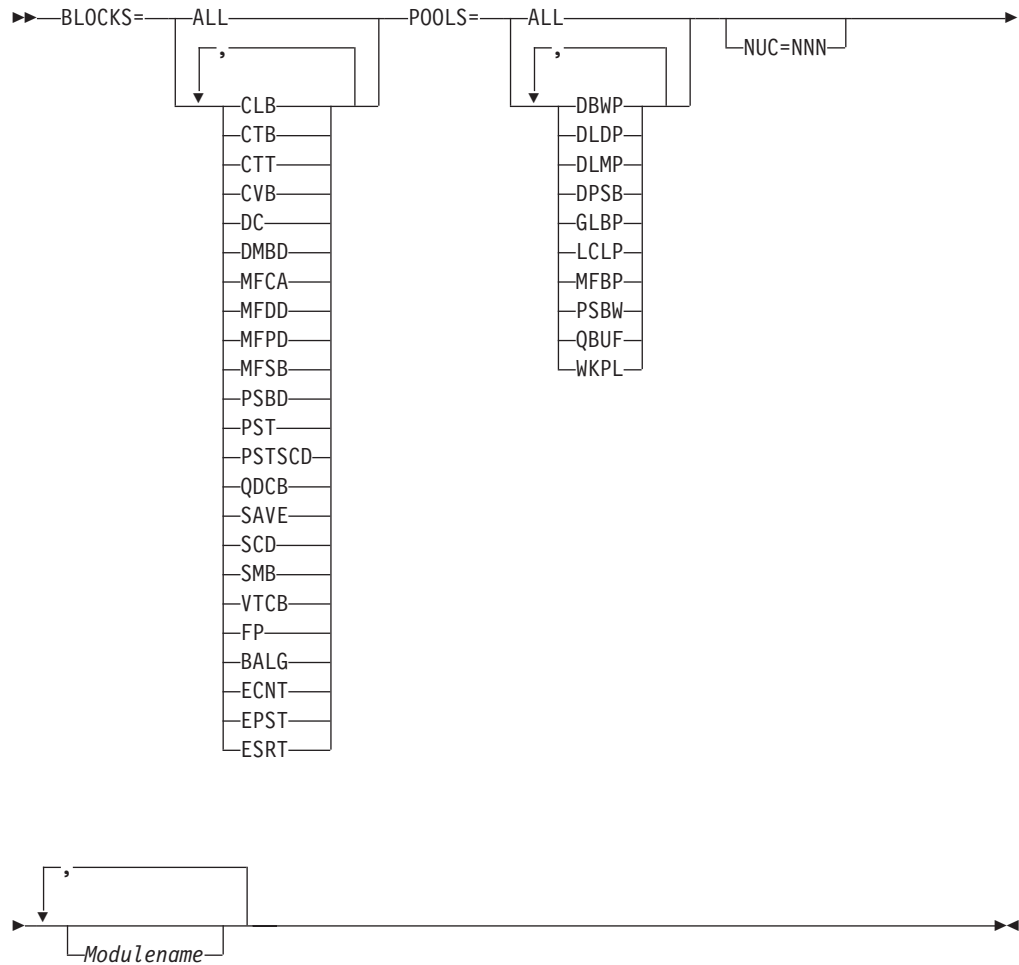
## **Member DFSFIXnn**

### **Defining Fixed Pages in the IMS Control Region**

You can tune your system to a particular operating environment by specifying that portions of the control region (for example, certain control blocks, buffer pools, loaded modules and part of the IMS nucleus) be fixed in address space during initialization. You must place control information in the PROCLIB procedure library (usually named IMS.PROCLIB), as part of member DFSFIXnn, where nn is a two-character field supplied in the PARM field of the EXEC statement for the control region. (See Chapter 4, "Procedures," on page 191.)

If you are running XRF, you can define an additional DFSFIXnn member. This member is identified by the DEFERFIX=xx parameter in the DFSHSBxx member of IMS.PROCLIB and can contain only non-Fast Path page-fix options for the control region. In the active system, both page-fix lists are processed. In the alternate system, the page fix list specified by the DEFERFIX=xx parameter is not processed until takeover. This allows you to control how much page fixing takes place in the alternate system while it is in tracking mode.

**Control Statements:** The control information is contained in 80-character records. Continuation or sequencing must not be entered. The format is as follows:



**Recommendation:** Specify BLOCKS=FP if your system has Fast Path specified by the FPCTRL macro. If you do not specify BLOCKS= either CPU performance can be seriously affected or system abends can be received, or both.

**BLOCKS=**

**ALL**

All blocks listed below.

**CLB<sup>1</sup>**

Non-VTAM communication line block

**CTB<sup>1</sup>**

Non-VTAM communication terminal block

**CTT<sup>1</sup>**

Communication translate table

**CVB**

Communication verb block

**DC<sup>1</sup>**

Non-VTAM CLB and non-VTAM CTB

**DMBD or DMBDIR**

Database directories

**MFCA**<sup>1</sup>

Message format buffer pool control area

**MFDD**<sup>1</sup>

MFS dynamic directory, (including hash table, prime area, and up to 10 additional areas)

**MFPD**<sup>1</sup>

MFS PDS directory indexes, (single area including one index per concatenation)

**MFSB**<sup>1</sup>

MFS staging buffers

**PSBD or PSBDIR**

Program directories

**PST**

Dependent region PSTs

**PSTSCD**

Blocks PST and SCD

**QDCB**<sup>1</sup>

Message queue DCBs

**SAVE**

Save area prefixes

**SCD**

System Contents Directory

**SMB**<sup>1</sup>

Scheduler message blocks

**VTCB**<sup>1</sup>

VTAM terminal control blocks

**FP**

BALG, ECNT, EPST, and ESRT

**BALG**<sup>1</sup>

Load balancing group control blocks

**ECNT**<sup>1</sup>

Extension to the Communication Node Tables (CNTs)

**EPST**

Extension to the Partition Specification Tables (PSTs)

**ESRT**<sup>1</sup>

Message retrieve buffers

**Note:** <sup>1</sup>This keyword is not applicable in the DBCTL environment.

The following Fast Path control blocks, which are always page fixed, are not included in the preceding list:

**BHDR**

Main storage database headers

**DMHR**

Database buffer headers

**LBUF**

Synchronization point log buffer

**DEDB**

Control blocks for data entry databases: data management control blocks (DMCBs) and data management area control blocks (DMACs)

**ESCD**

Extension to the System Content Directory (SCD)

**OTHR**

Output thread control blocks

**POOLS**

This is the dynamic area acquired by IMS during initialization and used for various buffer pools.

**ALL**

All pools listed below

**DBWP**

DMB work pool

**DLDP**

DMB pool

**DLMP**

With the DL/I address space option, that portion of the PSB pool in the z/OS common area.

**DPSB**

With the DL/I address space option, the portion of the PSB pool in DL/I local storage. Specification is ignored if the DL/I address space option is not in effect.

**GLBP**

Following storage pools are included: WKPL, DLMP, PSBW, DLDP, and DBWP. If the DL/I address space is used, DLDP and DBWP are not included.

**LCLP<sup>1</sup>**

Following storage pools are included: QBUF and MFBP. If the DL/I address space is used, DPSB, DLDP, and DBWP are also included.

**MFBP<sup>1</sup>**

Message format buffer pool

**PSBW**

PSB work pool

**QBUF<sup>1</sup>**

Message queue buffer pool

**WKPL**

General working pool

**Note:** <sup>1</sup>This keyword is not applicable in the DBCTL environment.

The parameters GLBP and LCLP can represent large amounts of storage. The terms GLBP and LCLP should not be interpreted as global and local storage, because the location of these pools can vary, based on whether the local storage option or the DL/I address space option is in effect. You should avoid using GLBP and LCLP; specify the pools to be fixed by name.

**NUC=NNN**

Is the percentage (specified by *nnn*) of the IMS load module (DFSVNUCx) to be page fixed.

If NUC=*nnn* is specified, a percentage of the nucleus, starting at location 0, is fixed regardless of what is placed there. IMS system definition places all the DC control blocks at the low end of the nucleus.

**Modulename**

The name of a loaded module. The specification of a module name does not cause that module to be fixed unless it has already been loaded into storage. (Module DFSVNUC0 cannot be fixed in this manner. See the NUC= keyword.)

If the DL/I address space option is being used, an attempt is made to fix a module in both the control and DL/I address spaces. If the module is loaded in only one of these address spaces, a message indicating that the module cannot be fixed is issued, in addition to the successful fix message.

**Fix Syntax Rules:**

- Either commas or blanks can be used to separate parameters, except BLOCKS, POOLS, and NUC, which are followed by an equal (=) sign.
- The *nnn* subparameter of NUC= can be any one, two, or three-digit number from 1 to 100.
- Following the BLOCKS or POOLS parameters, either ALL or any subset of the list of blocks or pools, respectively, can be coded.
- Parameters (NUC, BLOCKS, POOLS, Modulename) can be entered in any order and can be repeated as desired. If NUC is repeated, the percentage specified on the last NUC= is used. If BLOCKS or POOLS is repeated, a block or pool is fixed if it is specified in any of the BLOCKS= or POOLS= specifications.
- The subparameters of blocks and pools can be specified in any order.
- Invalid parameters are ignored, an error message is issued, and scanning continues.
- An invalid format or an invalid word (one not beginning with an alphabetic character) causes all scanning to stop and an error message to be issued. No fixing is done.
- All 80 characters in each record are available for control information. Continuation or sequencing cannot be entered.
- The modulename must be specified as the full eight-character name of a loaded module.

**Example:**

```
DFSDLR00 NUC=76 BLOCKS=DC,SMB,PSTSCD POOLS=MFBP
```

This example causes:

- Module DFSDLR00 to be fixed
- 76% of the IMS nucleus to be fixed
- The control program nucleus blocks DC, SMB, PST, and SCD to be fixed
- The message format buffer pool to be fixed

The member specified in the EXEC PARM field is read and processed during the IMS initialization procedure.

A list of the fixed pages is maintained during the execution of IMS. When IMS terminates, this list is used to free the fixed pages prior to returning to z/OS.

## Member DFSHSBxx

This member is not applicable in the DBCTL environment.

### Specifying the XRF Options

The DFSHSBxx member of IMS.PROCLIB specifies the XRF options used by the active and alternate subsystems in an XRF complex. Multiple DFSHSBxx members can be present in IMS.PROCLIB. The HSBMBR= parameter in the IMS procedure identifies the member currently in effect.

An XRF complex has two DFSHSBxx members in effect at any time, one for the active system, and one for the alternate subsystem. The two members need not be identical, but the USERVAR and RSENAME parameters must be the same, and the surveillance options should be the same. If MNPS is used for terminal switching, then the USERVAR parameter is not applicable. Instead, the MNPS parameters must be the same in the two members.

The parameters in this member can have different meanings, depending on whether the subsystem is the active or the alternate subsystem.

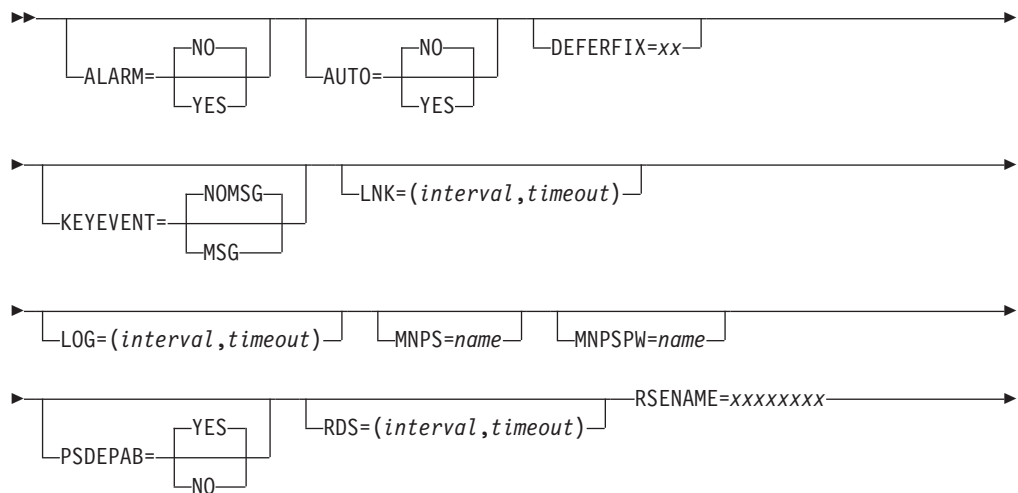
**Control Statement:** The record format is:

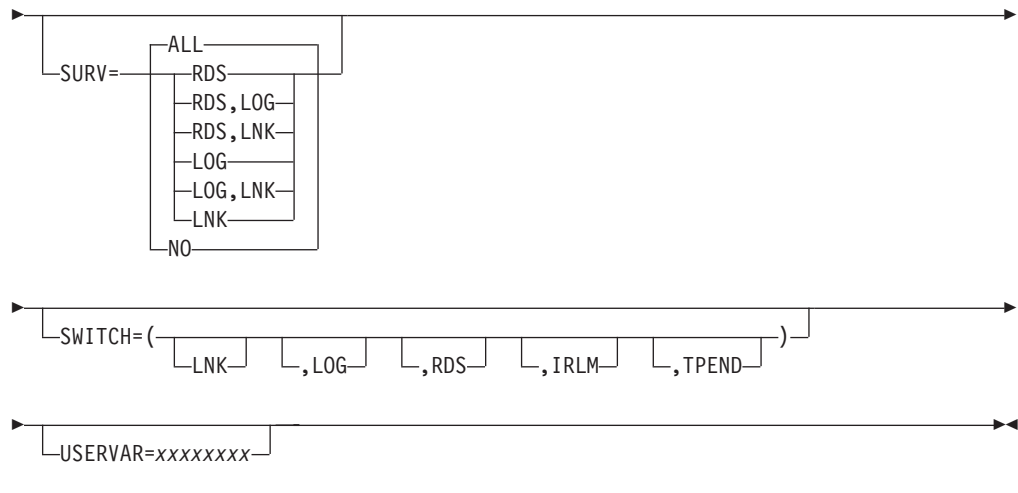
Position	Contents
1-71	Option parameters separated by commas
72-80	Ignored

The first parameter in a record can have leading blanks. The last parameter in a record is denoted by a comma, followed by one or more blanks. The last parameter in the last record is followed by one or more blanks.

**XRF Option Parameters:** The following XRF option parameters can occur in any order in the control statement. Multiple control statements can be provided.

RSENAME= is required for all XRF complexes. Either USERVAR= or MNPS= is required if you use VTAM for XRF terminal switching.



**ALARM=**

Specifies whether (YES) or not (NO) the service processor alarm on the alternate subsystem sounds at takeover request. The default is NO.

**AUTO=**

Specifies whether (YES) or not (NO) a takeover is to proceed automatically. The default is NO.

**DEFERFIX=xx**

Identifies the DFSFIXxx member of IMS.PROCLIB containing the non-Fast Path page fixing options for the control region, which are processed during the restart of an active subsystem, and when the alternate subsystem becomes the active during takeover.

**KEYEVENT=**

Controls whether your IMS operator receives all XRF- related messages.

- MSG indicates that noncritical-event messages and critical-event messages are sent.
- NOMSG indicates that only critical-event messages are sent.

The default is NOMSG.

**LNK=(interval,timeout)**

Specifies the timing values for the ISC link surveillance. The default is (3,9).

**interval**

In the active subsystem, specifies how often (1 to 99 seconds) IMS sends signals across the ISC link.

In the alternate subsystem, specifies how often (1 to 99 seconds) IMS checks for those signals.

See Table 41 on page 359 for additional information on coding the interval value.

**timeout**

In the alternate subsystem, specifies how long (1 to 999 seconds) IMS waits for a signal before considering a takeover.

See Table 42 on page 359 for additional information on coding the timeout value.



**LOG=(interval,timeout)**

Specifies the timing values for the log surveillance. It also determines the frequency with which the alternate subsystem accesses the log, after it has caught up. The default is (1,3).

**interval**

In the alternate subsystem, specifies how often (1 to 99 seconds) IMS checks to see if a new log record is received from the active subsystem.

This parameter also determines the frequency with which the alternate subsystem attempts to read a new log record, after the alternate has caught up with the active. If you do not specify LOG surveillance or do not specify the interval value, the alternate subsystem waits one second between reads to the log (only if the alternate has caught up with the active).

See Table 41 for additional information on coding the interval value.

Table 41. Initializing the INTERVAL Value

Subsystem	Requirement	Exception Action
Alternate	Alternate interval value must be greater than or equal to active interval value.	DFS3812 issued. Alternate interval value forced to active interval value.
Alternate	Alternate interval value must be greater than or equal to alternate LOG interval value (LNK and RDS).	DFS3833 issued. Alternate interval value forced to alternate log interval value.
Alternate	Alternate interval value less than or equal to alternate RDS and LNK interval value (LOG only).	DFS3833 issued. Alternate interval value forced to alternate RDS then LNK interval value.
Alternate	Twice the interval value must be less than or equal to the alternate timeout value.	DFS3832 issued. Alternate timeout value forced to twice the interval value.
Active	Twice the interval value must be less than or equal to the active timeout value.	DFS3832 issued. Active timeout value forced to twice the interval value.

**timeout**

In the alternate subsystem, specifies how long (1 to 999 seconds) IMS waits for a new record, before considering a takeover.

When the alternate subsystem catches up with the active, the timeout and interval values are also used to determine a delay interval, which limits the frequency of calls to DBRC to check if an OLDS switch has occurred.

See Table 42 for additional information on coding the timeout value.

Table 42. Initializing the TIMEOUT Value

Subsystem	Requirement	Exception Action
Alternate	Timeout value must be greater than or equal to twice the alternate interval value.	DFS3832 issued. Alternate timeout value forced to twice the alternate interval value.
Active	Timeout value must be greater than or equal to twice the active interval value.	DFS3832 issued. Active timeout value forced to twice the active interval value.

**MNPS=name**

Specifies the name of the MNPS ACB to use for terminal recovery. If specified,

the USERVAR specification is ignored. This keyword is optional. If this specification is omitted, IMS uses the traditional XRF support and the USERVAR= specification for recovery.

**MNPSPW=*name***

Specifies the password to use for the MNPS ACB. This password is checked by VTAM. If VTAM requires a password during VTAM system definition and no password is specified, then the MNPS ACB cannot be opened.

**PSDEPAB=**

Specifies whether (YES) or not (NO) prestarted BMP and IFP regions on the XRF alternate IMS are to be abnormally terminated with a U0454 abend during /ST0 BACKUP command processing on the alternate IMS, or /CHE FREEZE processing for the active IMS.

If NO is specified, the prestarted BMP and IFP regions are terminated with return code 0. The default is YES.

**RDS=(*interval,timeout*)**

Specifies the timing values for the RDS surveillance. The default is (1,3).

**interval**

In the active subsystem, specifies how often (1 to 99 seconds) IMS writes a timestamp on the RDS.

In the alternate subsystem, specifies how often (1 to 99 seconds) IMS checks for timestamps on the RDS.

See Table 41 on page 359 for additional information on coding the interval value.

**timeout**

In the alternate subsystem, specifies how long (1 to 999 seconds) IMS waits for a timestamp before considering a takeover.

See Table 42 on page 359 for additional information on coding the timeout value.

**RSENAME=*xxxxxxxx***

Specifies the name of the recoverable service element (RSE) containing the active and alternate IMS subsystems in an XRF complex. The RSENAME= specifications must be the same in DFSHSBxx members for the active and alternate subsystems.

The name used for the RSENAME parameter should not be the same name used for the IMSID parameter in the IMSCTRL macro for either the active or the alternate IMS subsystems in the XRF complex.

RSENAME= is required.

**SURV=**

Specifies the surveillance mechanisms to run in an XRF complex. The SURV= parameters in the DFSHSBxx members for the active and alternate subsystems must be the same. The default is RDS,LOG,LNK.

**RDS**

The alternate subsystem is to periodically check for timestamps in the RDS.

**LOG**

The alternate subsystem is to periodically check for new log records in the IMS log.

**LNK**

The alternate subsystem is to periodically check for signals coming across the ISC link.

**ALL**

All three surveillance mechanisms are to be used.

**SWITCH=**

Controls if the failure of a surveillance mechanism, VTAM, or the IRLM should cause the alternate subsystem to consider a takeover.

**LNK**

The absence of signals across the ISL link after a timeout period elapses causes IMS to consider a takeover.

**LOG**

The absence of a new log record in the IMS log after a timeout period elapses causes IMS to consider a takeover.

**RDS**

The absence of a new timestamp in the RDS after a timeout period elapses causes IMS to consider a takeover.

**IRLM**

An IRLM failure causes IMS to request a takeover.

**TPEND**

A VTAM failure that invokes the IMS TPEND exit routine causes IMS to request a takeover.

The default is SWITCH=(LNK,LOG,RDS),(IRLM),(TPEND), signifying IMS requests a takeover for any one of the following events.

- The LNK, LOG, and RDS surveillance mechanisms all indicate a failure.
- The IRLM fails.
- VTAM fails and causes the IMS TPEND exit routine to be invoked.

**USERVAR=xxxxxxx**

Specifies the USERVAR that VTAM uses to associate a logon message with the VTAM application name (APPLID) of the current active IMS subsystem in an XRF complex. The USERVAR= in DFSHSBxx members for the active and alternate subsystem must be the same.

USERVAR= is required if you have VTAM and are not using MNPS for XRF terminal switching.

When MNPS=*name* is specified, the USERVAR specification is ignored, and IMS uses MNPS for XRF terminal switching.

**Member DFSINTxx****Specifying Dependent Region Preinitialization Routines**

The DFSINTxx member of IMS.PROCLIB is used by MPR, IFP, and BMP dependent regions (PREINIT= exec parameter) to identify the preinitialization modules to receive control prior to region initialization.

**Related Reading:** For information on coding these routines, see *IMS Version 9: Customization Guide*.

**Control Statement:** The record format is:

Position	Contents
----------	----------

- 1-71            Module names, separated by a comma  
 72-80           Ignored

A separate control statement for each module is required.



## Member DFSJVMAP

This member maps all the 8-byte or less uppercase IMS Java application names specified to IMS in one of the following ways with the OMVS path name for the IMS Java application '.class' file:

- LANG=JAVA,GPSB= parm on the APPLCTN macro
- LANG=JAVA,PSB= parm on the PSBGEN macro
- MBR= parm on the DFSJBP procedure

DFSJVMAP pertains to JBP and JMP regions only. The member name must be DFSJVMAP.

DFSJVMAP is a member contained in a data set concatenated in the PROCLIB DD. A sample member DFSJVMAP is supplied in the IMS sample library. This member is optional and will be read during dependent region application scheduling, thus, making this member dynamic. You do not need to bring down IMS when adding application name mappings to this member for the change(s) to take affect.

Comments are supported for this member and must begin with an asterisk (\*) in the first column. Each line in this member must not be longer than 72 bytes. There must be no space between the equal sign (=) and the beginning of the pathname.

### DFSJVMAP Example 1

In this example, APPLCTN GPSB=JAVAPL1 was specified during IMS gen and javapl1.class exists in '/usr/IMSjava/applications'. You would specify the following in DFSJVMAP:

```
*****
* Path name for JAVAPL1                               *
*****
```

```
JAVAPL1=/usr/IMSjava/applications/javapl1
```

or

```
JAVAPL1=javapl1
```

### DFSJVMAP Example 2

In this example, APPLCTN PSB=JAVAPL2 was specified during IMS system definition and its corresponding PSBGEN macro was specified as PSBGEN LANG=JAVA,PSBNAME=JAVAPL2 and javapl2.class exists in '/usr/IMSjava/applications'. But javapl2.java contains the package statement 'package IMSjava.applications;'. You would specify the following in DFSJVMAP:

```
*****
* Path name for JAVAPL2 *
*****
JAVAPL2=IMSjava/applications/javap12
```

For this to work, you must have coded the following required JVM option in the correct JVMOPMAS= member:

```
Dibm.jvm.shareable.application.class.path=/usr:...
```

## Member DFSJVMMS

This is a sample member that demonstrates how to specify JVM options for the standalone JVM for JBP regions. For information about the format of this member, see “JVMOPMAS=” on page 279.

## Member DFSJVMWK

This is a sample member that demonstrates how to specify JVM options for the worker JVM for JMP regions. For information about the format of this member, see “JVMOPWKR=” on page 280.

## Member DFSMPLxx

### Making High-Use Program Modules Resident

In addition to those modules automatically preloaded into the IMS control region, other z/OS, IMS, or user-written program modules can be made resident in IMS regions by using the IMS Module Preload function. Doing so can improve throughput and response time for transactions frequently referred to if sufficient virtual storage is available with high-performance paging DASD. Ordinarily, the Module Preload function is advantageous only for MPPs and for message-driven Fast Path regions, and only when the preloaded modules are reentrant.

Program modules can be made resident in either an IMS region or LINKPACK. This section explains how to make program modules permanently resident and provide some rules for doing so.

**Resident in a Region:** When making modules resident in a region, consider the following:

- Serially reusable z/OS, IMS, or user-written program modules only can be resident in a region.
- Only program modules used for transactions serviced by the region involved, should be made resident in that region. Program modules should be resident for only the duration of the region.
- z/OS and IMS modules that execute in a dependent region can reside in that dependent region.

Modules made resident in a region are in the region JOBPack and are called without repeating the overhead of searching STEPLIB/JOBLIB, LINKPACK, and SYS1.LINKLIB. The overhead of fetching the module into virtual storage is encountered only at region initialization time.

**Resident in LINKPACK:** In an operating environment where there are several batch regions or a combination of online and batch regions, it can be advantageous to place some of the frequently used IMS and access method modules in the operating system LPA. Programs made resident in LPA should be those that can be shared among all regions. This saves virtual storage space. The modules to be

included must exist in either a LNKSTxx member of SYS1.PARMLIB (or its concatenations, one of which should be IMS.SDFSRESL) or SYS1.SVCLIB.

Initial access of LPA resident program modules can be slow, because the region JOBPack and STEPLIB/JOBLIB are searched before LPA is searched. Subsequent access can be at CPU speeds if the region JOBPack has not been purged by z/OS virtual storage management (as when sufficient virtual storage is not available to satisfy a user GETMAIN for space). Although the modules are physically residing in LPA (and are shared among multiple regions), the overhead involved in searching program libraries and LPA is only experienced at region initialization time.

Although some real storage can be saved by putting the DL/I action modules into LPA, it is not recommended that an installation place IMS modules into the LPA. Doing so makes it difficult to execute different versions of IMS concurrently. Also, when maintenance is applied to modules in LPA, another initial program load of the z/OS system is required.

**The IMS Module Preload Function:** The z/OS task under which modules are preloaded varies based on the IMS region type. Table 43 shows IMS region types and associated z/OS tasks.

Table 43. IMS Region Types and Associated z/OS Tasks

IMS Region Type	z/OS Task
Control (CTL)	Physical Log
Message (MSG) non-reentrant modules	Program Control
Message MSG reentrant modules	Region Control
Batch Message (BMP)	Region/Program Control
Batch (DLI)	Region/Program Control
Fast Path (IFP)	Region/Program Control

**Control Statement:** Complete the following steps to use this function to make program modules resident in a region or LINKPACK.

1. Use the IMS.PROCLIB data set allocated prior to Stage 2 of IMS system definition and the z/OS utility IEBUPDTE to create IMS.PROCLIB members that identify modules to be preloaded.

**Related Reading:** See *MVS/ESA Data Administration: Utilities*. Member names must be DFSMPLxx. The record format is:

Position	Contents
1-71	Module names, separated by a comma
72-80	Ignored

The first name on a record can have leading blanks. The last name on a record is denoted by a comma followed by one or more blanks; the last name on the last record is followed by one or more blanks.

No limit exists for the number of module names that can be specified. If a preloaded module has ALIAS names that are ordinarily invoked via z/OS LINK (that is, application programs in DLI BMPs and MPPs), those ALIAS names should also be specified for preload.

2. Bind the application program modules using the IMS reentrant DL/I language interface. The true attributes must be specified; for example, if reusable only, do not specify "RENT."

The IMS DLI language interface is not reentrant. Any IMS application programs that were designed to be reentrant or serially reusable can use the module preload function after being bound with the IMS language interface.

3. In the step execution JCL, do both of the following:
  - Insert a DD statement for IMS.PROCLIB with the ddname PROCLIB.
  - Specify the correct PROCLIB member in the EXEC statement parameter.

If an invalid member name is specified, or if the record format is incorrect, no modules are loaded but a message is issued and initialization continues.

The member name should be unique unless overridden by the IMS procedure.

For a description of the EXEC statement parameters, see “DFSMPR Procedure” on page 219.

## Member DFSORSxx

This member applies to DBCTL or DB/DC online environments. The DFSORSxx member of the IMS.PROCLIB provides system-related startup parameters for the recovery manager. This member is identified by the ORSMBR parameter in the EXEC statement (see “Parameter Descriptions for IMS Supplied Procedures” on page 255 for a description of ORSMBR). If the IMS.PROCLIB member DFSORSxx is not provided, the installed recovery service is initiated with default values.

### Related Reading:

- For information about tailoring the DFSORSxx member for use with the IMS Online Recovery Service product, see *IMS Online Recovery for z/OS User's Guide*.
- For information about tailoring the DFSORSxx member for use with the IMS Database Recovery Facility V2 product, see *IMS Database Recovery Facility for z/OS User's Guide*.

## Member DFSPBxxx

The three variations of DFSPBxxx (DFSPBDBC, DFSPBDCC, and DFSPBIMS) allow you to specify execution parameters for the control region. Use a particular DFSPBxxx member for different IMS environments.

### **DFSPBDBC**

DBCTL environment

### **DFSPBDCC**

DCCTL environment

### **DFSPBIMS**

DB/DC environment

You can use the IMS Syntax Checker to modify these PROCLIB members. For more information, see “IMS Syntax Checker” on page 332.

The DFSPBxxx member consists of one or more 80-character records. Each record contains either:

- A comment, marked by an '\*' in statement column 1.
- One or more execution parameters in the following form:

KEYWORD=value

The parameters are separated by commas (.). Each parameter can be preceded by a space or the comma following the prior parameter value. You can have blanks on either, both, or neither side of the comma that separates keyword strings.

- One execution parameter

KEYWORD=value

followed by a comment. The comment is separated from the parameter by one or more blanks. A comment may contain any characters except the equal sign (=).

An example of the DFSPBxxx member is shown below. For a list of parameters and parameter descriptions, see "IMS Procedure" on page 231.

```
ARC=,
AUTO=,
```

### **DFSPBDBC**

This member applies only to a DBCTL environment. It allows you to specify DBCTL control region execution parameters. The parameters that you specify in this member override those specified in the stage 1 macros.

You can place several DFSPBDBC members in PROCLIB by replacing the member name DFSPBDBC with DFSPBxxx, where xxx must be three alphanumeric characters. The RGSUF= keyword in the DBC procedure specifies the xxx suffix to be used during startup of the DBCTL control region.

### **DFSPBDCC**

This member applies only to a DCCTL environment. It allows you to specify DCCTL control region execution parameters. The parameters that you specify in this member override those specified in the stage 1 macros.

You can place several DFSPBDCC members in PROCLIB by replacing the member name DFSPBDCC with DFSPBxxx, where xxx must be three alphanumeric characters. The RGSUF= keyword of the DCC procedure provides the xxx suffix to be used during startup of the DCCTL control region.

### **DFSPBIMS**

This member applies only to a DB/DC environment. It allows you to specify DB/DC control region execution parameters. The parameters that you specify in this member override those specified in the stage 1 macros.

You can place several DFSPBIMS members in PROCLIB by replacing the member name DFSPBIMS with DFSPBxxx, where xxx must be three alphanumeric characters. The RGSUF= keyword of the IMS procedure provides the xxx suffix to be used during startup of the DB/DC control region.

## **Member DFSRSRxx**

### **Specifying the RSR Options**

The DFSRSRxx member of IMS.PROCLIB specifies the RSR options used by the online active and tracking subsystems in an RSR complex. Multiple DFSRSRxx members can be present in IMS.PROCLIB. The RSRMBR= parameter in the IMS, DBC, and DCC procedures identifies the member currently in effect. IMS batch jobs do not use the DFSRSRxx member.



If the RSR feature is installed and you do not want to use it, you must override RSR enablement by specifying RSR(NO) in the DFSRSRxx proclib member. For more information, see member DFSRSRxx.

An RSR complex has at least two DFSRSRxx members in effect at any time, one for the online active system and one for the tracking subsystem. The two members need not be identical, but the GSGNAME parameters must be the same. The parameters can have different meanings, depending on whether the subsystem is the active or the tracker.

If RSRFEAT=RLT or RSRFEAT=DLT is not specified in the IMSCTRL macro during IMS system definition, the specification of a GSGNAME and TMINAME in the DFSRSRxx member is not accepted.

**Control Statement:** Because the parameters in the DFSRSRxx member are parsed by the TSO parser, the parameters must follow TSO parameter syntax rules. Any parsing errors are recorded in the SYSTSPRT data set. The record format is:

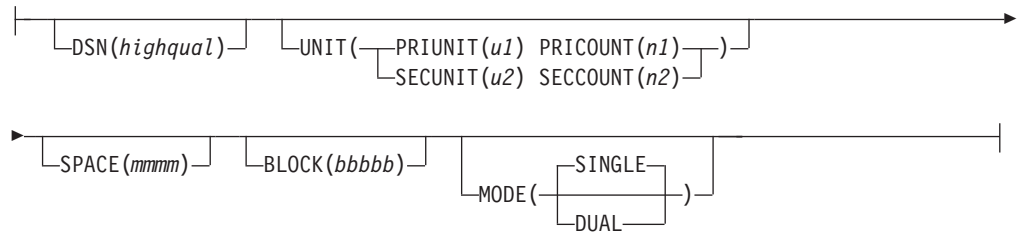
Position	Contents
1-80	Option parameters separated by commas

The first parameter in a record can have leading blanks. Delimiters between parameters can be spaces or commas. The last parameter in a record is denoted by a comma, followed by one or more blanks. The last parameter in the last record is followed by one or more blanks. Comments are delimited by /\* and \*/.

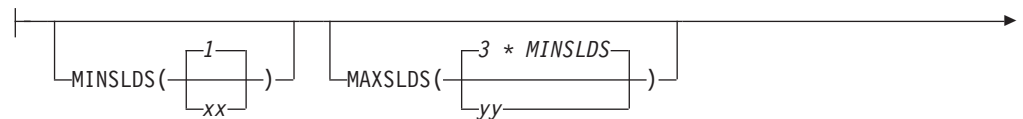
**RSR Option Parameters:** All of the RSR option parameters can occur in any order in the control statement. Multiple control statements can be provided.

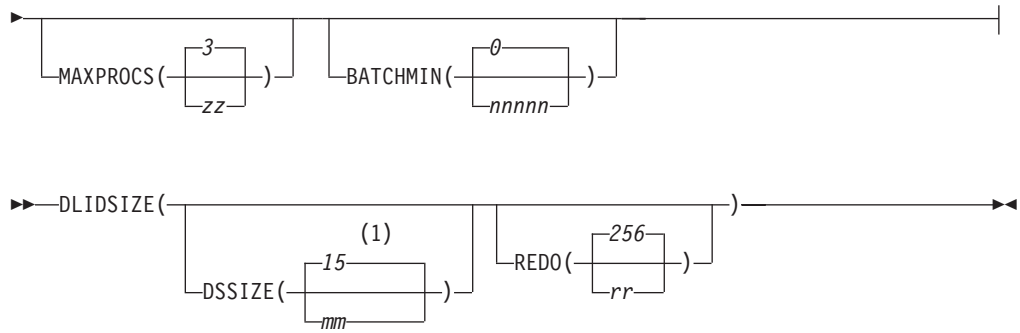


**AA:**



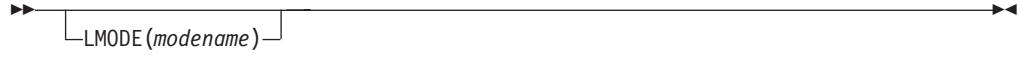
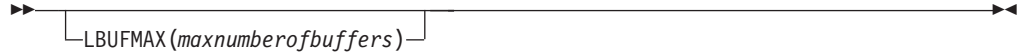
**BB:**



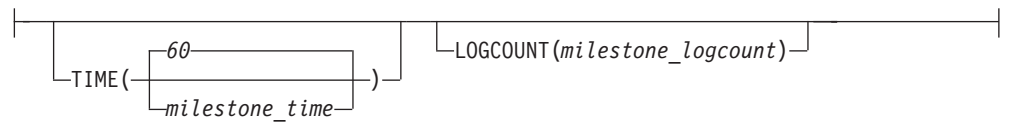


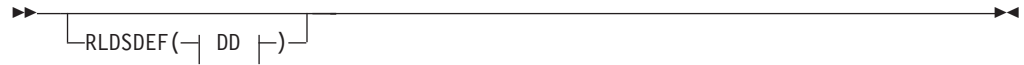
**Notes:**

- 1 Or the number of PSTs, whichever is greater.

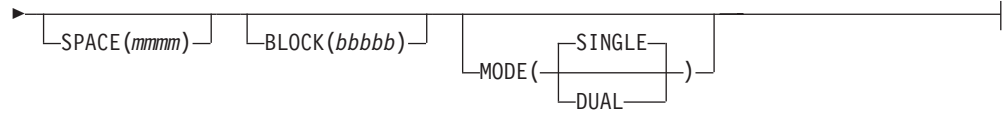
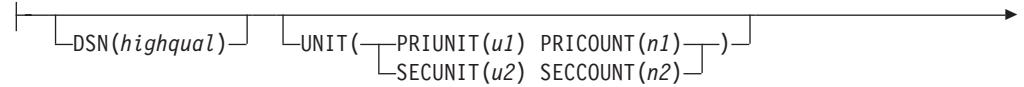


**CC:**

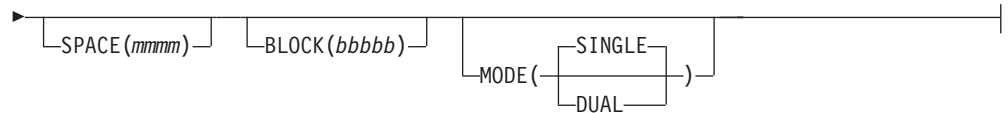
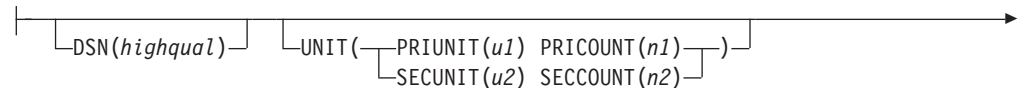




**DD:**



**EE:**



**ARCHDEF=**

Defines the archived secondary system log data set (SLDS) characteristics for the tracking subsystem.

If ARCHDEF is not specified, no automatic archiving occurs.

**DSN**

Specifies the initial qualifiers of the data set name to be used for archive data sets.

No default data set name exists.

The character string specified must satisfy data set name syntax rules and be less than or equal to 27 characters in length. The actual data set name generated looks like: highqua1.ARCHc.Nnnnnnn, where c is 1 (one) for the primary SLDS archive and 2 (two) for the secondary SLDS archive, and nnnnnnn is a number generated to identify uniquely the data set name.

**UNIT=**

Specifies the primary and secondary unit types to be used when creating or reading SLDS archive data sets.

**PRIUNIT=**

Specifies the unit type to be used for the data set (if MODE(SINGLE) is specified) or for the primary data set (if MODE(DUAL) is specified). Example unit types are: SYSDA, TAPE, 3380, 3390.

No default unit type exists. You cannot specify different device types for the two units used in dual logging of a tracker log data set type.

**PRICOUNT=**

Specifies the number of devices to be allocated for the data set or primary data set. This is most meaningful for tape devices, because multiple tapes can be mounted, thus eliminating the need to wait for tape rewind and new tape mount.

The default for PRICOUNT is zero. Zero specifies a data set consisting of up to five volumes for tape data sets. Specifying any other value for PRICOUNT specifies multi-volume data sets in a maximum of 255 volumes per data set.

**Recommendation:** Specify PRICOUNT(0) and SECCOUNT(0) to limit the number of volumes to five or less when allocating archived SLDS during restart. Ensure the amount of data being archived will fit into five volumes if tape data sets are being used.

**SECUNIT=**

Specifies the unit type to be used for the secondary data set. Example unit types are: SYSDA, TAPE, 3380, 3390.

No default unit type exists. You cannot specify different device types for the two units used in dual logging of a tracker log data set type.

**SECCOUNT=**

Specifies the number of devices to be allocated for the secondary data set. This is most meaningful for tape devices, because multiple tapes can be mounted, thus eliminating the need to wait for tape rewind and new tape mount.

The default for SECCOUNT is 0 (zero). A specification of 0 results in data sets consisting of up to five volumes for tape data sets. Any other value for SECCOUNT results in a maximum of 255 volumes per data set.

**SPACE=**

Specifies the space allocation in megabytes.

No default space allocation exists.

For DASD, the value specified is converted to blocks to give an integral number of cylinders. For tape units, the value specified controls switching from one data set to the next.

For tape units, if your space allocation is small enough to cause tape end-of-volume before the physical end of the tape, the data set overflows to

another volume. The space allocation must be greater than the product of the values specified in SLDSDEF SPACE and ARCHIVE MINSLDS.

((SLDSDEF SPACE) \* MINSLDS <= (ARCHDEF SPACE))

### **BLOCK=**

Specifies the block size to be used for all SLDS archive data sets. No default block size exists.

This number should be a fairly large number to ensure good RSR performance. A block size of 32 KB is recommended for tape and half-track for DASD (22 KB for 3380 and 26 KB for 3390 devices). The block size must be at least as large as the largest incoming IMS log record. The range of valid values is 4096 to 32760.

### **MODE**

Specifies whether SINGLE or DUAL archive SLDS logging is required. The default is SINGLE.

### **ARCHIVE**

Controls automatic archiving of SLDS data sets created by a tracking subsystem. If ARCHIVE is not specified, no automatic archiving occurs.

**Related Reading:** See *IMS Version 9: Operations Guide* for more information on how to use ARCHIVE to control automatic archiving of tracking SLDSs.

### **MINSLDS**

Specifies the minimum number of tracking SLDSs to be archived for each active subsystem.

The range of valid values is 1 to 999. The default is 1.

### **MAXSLDS**

Specifies the maximum number of tracking SLDSs to be maintained for each active subsystem before an archive operation will be triggered.

The range of valid values is the MINSLDS value to 999. The default is 3 times MINSLDS.

### **MAXPROCS**

Controls the number of concurrent archive operations.

The range of valid values is 1 to 99. The default is 3.

### **BATCHMIN**

Specifies the minimum number of log records required before the system initiates automatic archiving. BATCHMIN is valid for log records that originate from active site batch IMS jobs. If a batch IMS job generates fewer log record than specified by the BATCHMIN parameter, the tracking SLDS is not archived.

The range of valid values is 1 to 999999.

If BATCHMIN is not specified, all tracking SLDS from batch IMS jobs are automatically archived when automatic archive is active.

### **DLIDSIZE=**

Specifies how the DL/I database tracker uses its data space. The specification indicates the amount of virtual storage for the DL/I tracking data space that can be backed by real and expanded storage for those times when the database tracker cannot keep up with the active subsystems.

If you specify a number larger than the available storage for the database tracker, the machine can experience excessive paging, thus reducing the efficiency of the tracker. If you specify a number smaller than the available

storage for the database tracker, the tracker might need to perform catch-up processing more often than is truly necessary.

**DSSIZE=**

Specifies the maximum number of megabytes for DL/I database tracking data space usage.

The range of valid values is 15 to 1600. The default is either 15 or the number of PSTs (set by the PST parameter on the EXEC PARM= statement), whichever is greater.

**REDO=**

Specifies an estimated average size, in bytes, for the active subsystem's DL/I redo (type X'50') log records. The number specified is rounded up to the nearest power of two.

The range of valid values is 128 to 4096. The default is 256.

**FPDSSIZE**

Specifies the threshold value size of storage, in megabytes, used for Fast Path database tracking data spaces. When data space usage reaches the value specified, the stored update data is written to disk. Actual storage size can be larger than the specified value, because additional data is normally processed while data is written to disk.

The range of valid values is 1 to 1024. The default is 1024.

**GSGNAME=**

Specifies the one- to eight-character global service group name to be used for the particular RSR complex. This GSG name overrides any specified during system definition (in the IMSCTRL macro).

Both the active and tracking subsystems must specify the same GSG name, either during system definition or in the DFSRSRxx member.

No default GSG name exists.

**ILTMODE=**

Specifies the one- to eight-character mode name to be used by the transport manager subsystem for isolated log sender conversations.

If ILTMODE is not specified, the default logon mode name defined in the DLOGMOD of the transport manager subsystem's VTAM APPL definition is used.

**LBUFMAX=**

Specifies the maximum number of buffers that the active Transport Manager subsystem will obtain for transmitting log data. Transport Manager buffers are 52 bytes larger than IMS log buffers and are obtained from IMS private storage. When LBUFMAX is not specified or cannot be specified (such as with IMS batch jobs), a limit of ten times the number of IMS log buffers is used.

**Restriction:** The LBUFMAX parameter only applies to the active system.

**LMODE=**

Specifies the one- to eight-character mode name to be used by the transport manager subsystem for the logger and log router conversations.

If LMODE is not specified, the TMS-implementation defined default logon mode name, TMDEFLT, is used.

**MILEINTV=**

Specifies the interval between milestones for tracking subsystems. Milestones are used by the tracking subsystem for tracker restart. Tracker restart is not a

remote takeover. They keep track of the current positions on the SLDSs containing active IMS log records and of which log records have been applied to the shadow databases.

The milestone can be triggered either by time interval or by number of log records received from all active subsystems. You can specify either or both of these methods for triggering milestones, but the timer (default or specified value) is used in all cases. If both are specified, the first one reached (timer or log record count) triggers the milestone and both counters are reset to zero.

**TIME=**

Specifies the maximum elapsed time between milestones. The range of valid values is 10 to 600 seconds. The default is 60 seconds.

**LOGCOUNT=**

Specifies the maximum amount of log data that can be received from all active subsystems between milestones.

The value can be specified as either *nnnK* (indicating the number of records of data in thousands) or as *nnnM* (indicating the number of records of data in millions). The range of valid values is 1K to 2047M (or 2096128K) records. No default milestone log count exists.

**MTOID=**

Specifies which terminals to use for the IMS master terminal and secondary master terminal. The terminals specified must be listed on the **NAME=** parameter of the **TERMINAL** macro. **MTOID** also controls which password is used. The password must be included on the list in the **PASSWD=** parameter of the **COMM** macro.

This parameter is only valid for tracking subsystems.

Valid values are 1, 2 or 3, depending on the number of names specified in the **TERMINAL** and **COMM** macros. The default is 1. The VTAM **APPLID** is overridden by the **APPLIDn=** startup parameter and **APPLID3** is always used for the tracking subsystem.

You must take care not to cause the same IMS VTAM **APPLID** to be opened for more than one IMS subsystem. If necessary, the **APPLID1**, **APPLID2** and **APPLID3** parameters should all be overridden to ensure unique **APPLIDs**.

**RETPD=**

Specifies the retention period, *nnnn*, for a tracking log data set, where *nnnn* is a one- to four-digit value representing a number of days. The system adds the number you specify to the current date to arrive at an expiration date. If you do not specify **RETPD**, the default is either the system default for this class of data set or 0. Use of this optional parameter helps reduce the possibility of accidental deletion of the tracking log data set. After the retention period expires, another data set can write over the tracking log data set. You cannot use the **RETPD** parameter to change the expiration date of existing tracking log data sets if they are managed by SMS.

You can also specify the retention period of the tracking log data set using the **EXPDT** parameter on standard JCL. Code the **RETPD** parameter when you want to specify a retention period for the tracking log data set or when you want to override the retention period defined in the data class for new tracking log data sets.

The **RETPD** parameter cannot have a null value.

**RLDSDEF=**

Defines the recovery log data set (RLDS) characteristics for the tracking subsystem.

No default RLDS definition exists. If RLDSDEF is not specified, no RLDS is created at the tracking site.

**DSN=**

Specifies the initial qualifiers of the data set name to be used for RLDS data sets.

No default data set name exists.

The character string specified must satisfy data set name syntax and be less than or equal to 27 characters in length. The actual data set name generated looks like: highqual.RLDS.c.Nnnnnnn, where c is 1 for the primary RLDS data set and 2 for the secondary RLDS, and nnnnnnn is a number generated to uniquely identify the data set name.

**UNIT=**

Specifies the primary and secondary unit types to be used when creating or reading RLDS data sets.

**PRIUNIT=**

Specifies the unit type to be used for the data set (if MODE(SINGLE) is specified) or for the primary data set (if MODE(DUAL) is specified). Example unit types are: SYSDA, TAPE, 3380, 3390.

No default unit type exists. You cannot specify different device types for the two units used in dual logging of a tracker log data set type.

**PRICOUNT=**

Specifies the number of devices to be allocated for the data set or primary data set. Specifying PRICOUNT is most useful for tape devices, so that multiple tapes can be mounted, thus eliminating the need to wait for tape rewind and new tape mount.

The default for PRICOUNT is 0 (zero). A specification of 0 results in data sets consisting of up to five volumes for tape data sets. Any other value for SECCOUNT results in a maximum of 255 volumes per data set.

**Recommendation:** Specify PRICOUNT(0) and SECCOUNT(0) to limit the number of volumes to five or less when allocating archived RLDS during restart. Ensure the amount of data being archived will fit into five volumes if tape data sets are being used.

**SECUNIT=**

Specifies the unit type to be used for the secondary data set. Example unit types are: SYSDA, TAPE, 3380, 3390.

There is no default unit type. You cannot specify different device types for the two units used in dual logging of a tracker log data set type.

**SECCOUNT=**

Specifies the number of devices to be allocated for the secondary data set. This is most meaningful for tape devices because multiple tapes can be mounted, thus eliminating the need to wait for tape rewind and new tape mount.



The default for SECCOUNT is zero. A specification of zero results in data sets consisting of up to five volumes for tape data sets. Any other value for SECCOUNT results in a maximum of 255 volumes per data set.

**SPACE=**

Specifies the space allocation, in megabytes, for each RLDS data set.

The range of valid values is 1 to 4095. No default space allocation exists.

**BLOCK=**

Specifies the block size to be used for all RLDS data sets. No default block size exists.

This value should be a fairly large number to ensure good RSR performance. A block size of 32 KB is recommended for tape archiving and half-track for DASD (22KB for 3380 and 26KB for 3390 devices). The block size must be at least as large as the largest incoming IMS log record. The range of valid values is 4096 to 32760.

**MODE=**

Specifies whether SINGLE or DUAL RLDS logging is required. The default is SINGLE.

**RSR=**

Specifies whether the active IMS subsystem is to be enabled for RSR processing. The default is YES.

With RSR(YES), you must also specify a GSG name and a TMI name, either in the DFSRSRxx member or in the IMSCTRL macro

**Important:** If the RSR feature (RLT, DLT, or both) is installed and the GSGNAME keyword is specified on the IMSCTRL macro, but you do not want RSR enabled for this particular IMS, you **must** specify RSR(NO). See the keyword descriptions under "IMSCTRL Macro" on page 87 for information about the GSGNAME keyword.

**SLDSDEF=**

Defines the shadow system log data set (SLDS) characteristics. This parameter is required for all tracking subsystems. No default for SLDS definition exists.

**DSN=**

Specifies the initial qualifiers of the data set name to be used for SLDS data sets. No default data set name exists.

The character string specified must satisfy data set naming conventions and be less than or equal to 27 characters in length. The actual data set name generated looks like: highqual.SLDS<sub>c</sub>.Nnnnnnnn, where c is 1 for the primary SLDS data set and 2 for the secondary SLDS, and nnnnnnn is a number generated to uniquely identify the data set name.

**UNIT=**

Specifies the primary and secondary unit types to be used when creating or reading SLDS data sets.

**PRIUNIT=**

Specifies the unit type to be used for the data set (if MODE(SINGLE) is specified) or for the primary data set (if MODE(DUAL) is specified). Example unit types are: SYSDA, TAPE, 3380, 3390.

No default unit type exists. You cannot specify different device types for the two units used in dual logging of a tracker log data set type.

**PRICOUNT=**

Specifies the number of devices to be allocated for the data set or primary data set. This is most useful for tape devices because multiple tapes can be mounted, thus eliminating the need to wait for tape rewind and new tape mount.

The default for PRICOUNT is 0 (zero). Specifying 0 results in data sets consisting of up to five volumes for tape data sets. Any other value for PRICOUNT results in a maximum of 255 volumes per data set.

**SECUNIT=**

Specifies the unit type to be used for the secondary data set. Example unit types are: SYSDA, TAPE, 3380, 3390.

No default unit type exists. You cannot specify different device types for the two units used in dual logging of a tracker log data set type.

**SECCOUNT=**

Specifies the number of devices to be allocated for the secondary data set. This is most useful for tape devices, because multiple tapes can be mounted, thus eliminating the need to wait for tape rewind and new tape mount.

The default for SECCOUNT is 0 (zero). Specifying 0 results in data sets consisting of up to five volumes for tape data sets. Any other value for SECCOUNT results in a maximum of 255 volumes per data set.

**Recommendation:** Specify PRICOUNT(0) and SECCOUNT(0) to limit the number of volumes to five or less when allocating archived SLDS during restart. Ensure the amount of data being archived will fit into five volumes if tape data sets are being used.

If you are using database readiness level on your tracking subsystem, the recommendation is that the UNIT be a direct access device, because the volume mounting characteristics of tape devices can result in unsatisfactory operation.

**SPACE=**

Specifies the space allocation, in megabytes, for each SLDS data set.

The range of valid values is 1 to 4095. No default space allocation exists.

**BLOCK=**

Specifies the block size to be used for all SLDS data sets. No default block size exists.

This value should be a fairly large number to ensure good RSR performance. A block size of 32 KB is recommended for tape archiving and half-track for DASD (22KB for 3380 and 26 KB for 3390 devices). The block size must be at least as large as the largest incoming IMS log record. The range of valid values is 4096 to 32767.

**MODE=**

Specifies whether SINGLE or DUAL SLDS logging is required. The default is SINGLE.

**TMINAME=**

Specifies the one- to four-character transport manager instance name that is used to identify a particular transport manager subsystem in the RSR complex.

If specified, TMINAME overrides the TMINAME= parameter in the IMSCTRL macro.

If no *tminame* is specified, IMS will use a series of blanks as the default *tminame*, and associates IMS with the TMS that uses a default instance name.

#### **TRKMODS=**

Specifies the tracking subsystem's MODSTAT DFSMDA member names for tracking MODSTAT status of the active subsystem.

This parameter is only valid for active IMS subsystems, and use of the parameter prohibits sharing of DFSRSRxx members between active subsystems.

The DD names specified must be included in the dynamic allocation list at the tracking site, and the data sets defined by these DD names need to be available to the tracking subsystem.

A tracking subsystem that was previously an active subsystem is not required to use the same DD name as was specified for it when it was an active subsystem.

#### **MODSTAT=**

Specifies the DFSMDA member name used to track the MODSTAT data set.

This parameter is only valid for IMS active subsystems.

No default member name exists. If MODSTAT is not specified, no MODSTAT tracking is done.

#### **MODSTAT2=**

Specifies the DFSMDA member name used to track the MODSTAT2 data set.

This parameter is only valid for XRF alternate subsystems.

No default member name exists. If MODSTAT2 is not specified, no MODSTAT2 tracking occurs.

## **Member DFSSPMxx**

This member allows you to override the default buffer definitions for the storage pools managed by the DFSPPOOL storage manager.

Default storage pool definitions are generated by IMS. Each definition contains information on the pool including the default buffer definitions.

The buffer size definitions contain the buffer sizes the Storage Manager is allowed to choose from when allocating a buffer from the pool. For each buffer size, the definition specifies how many buffers to obtain in the primary block, how many buffers to obtain in secondary blocks, and whether or not to obtain the primary block when the pool is allocated. If the primary block is obtained during initialization, it is not released during compression.

The buffer definitions shown in Figure 32 on page 378 are the existing default definitions.

For all pools, 16 bytes are added to the buffer sizes for prefix and suffix information. For CIOP, HIOP, EMHB, LUMP, LUMC, and SPAP pools, an additional eight bytes are added for the overlay detection constant.

For EMHB, the buffer size includes the prefix and data portion. For information on the prefix length, see the EMHBHL field of the expedited message handler block control block that is mapped by the DBFEMHB macro.

AOIP								
	BUF1	BUF2	BUF3	BUF4	BUF5	BUF6	BUF7	BUF8
SIZE	48	132	256	576	1048	2096	4192	32768
PRIM. ALLOC.	50	500	100	32	16	8	4	4
SEC. ALLOC.	50	1500	100	32	8	4	2	2
INIT.	Y	N	N	N	N	N	N	N

CIOP								
	BUF1	BUF2	BUF3	BUF4	BUF5	BUF6	BUF7	BUF8
SIZE	256	512	1024	2048	4096	8192	16384	32768
PRIM. ALLOC.	64	32	32	32	16	8	4	4
SEC. ALLOC.	32	16	16	16	8	4	2	2
INIT.	Y	N	N	N	N	N	N	N

HIOP, EMHB								
	BUF1	BUF2	BUF3	BUF4	BUF5	BUF6	BUF7	BUF8
SIZE	256	512	1024	2048	4096	8192	16384	32768
PRIM. ALLOC.	64	64	32	32	16	8	4	4
SEC. ALLOC.	32	32	16	16	8	4	2	2
INIT.	N	N	N	N	N	N	N	N

CESS								
	BUF1	BUF2	BUF3	BUF4	BUF5	BUF6	BUF7	BUF8
SIZE	128	256	512	1024	2048	4096	8192	16384
PRIM. ALLOC.	32	32	32	32	32	16	8	4
SEC. ALLOC.	16	16	16	16	16	8	4	2
INIT.	Y	N	N	N	N	N	N	N

FPWP								
	BUF	BUF	BUF	BUF4	BUF5	BUF6	BUF7	BUF8
SIZE	25	51	102	2048	4096	8192	16384	32768
PRIM. ALLOC	6	6	3	32	16	8	4	4
SEC. ALLOC.	3	3	1	16	8	4	2	2
INIT.				N	N	N	N	N

LUMP, LUMC								
	BUF1	BUF2	BUF3	BUF4	BUF5	BUF6	BUF7	BUF8
SIZE	128	256	512	1024	2048	3064	4096	33024
PRIM. ALLOC.	32	32	32	32	16	12	8	4
SEC. ALLOC.	32	16	16	16	8	12	8	2
INIT.	N	N	N	N	N	N	N	N

Figure 32. Default Buffer Definitions

With DFSSPMxx, you can override buffer definitions for any of the pools identified in Figure 32:

AOIP  
CESS

CIOP  
 EMHB  
 FPWP  
 HIOP  
 LUMC  
 LUMP

**Exception:** If you establish an FDBR region, all pool names that you specify are ignored. The FDBR region internally specifies the following values for all pools regardless of whether they are explicitly specified by the control statements or defined by SYSGEN as default:

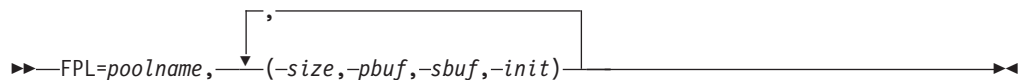
- 2 Number of buffers in the primary storage allocation
- 2 Number of buffers in secondary storage allocation
- N Primary storage allocation during IMS initialization =

You can define a maximum of thirty-two different buffer sizes that can be used to satisfy requests for storage. The size of the buffer that is used to satisfy a request is determined on a “best-fit” basis.

The FPL= statement is used to specify buffer definition overrides. Multiple buffer definitions are allowed on a single statement. You can have one or more FPL= statements within the DFSSPMxx member. If the member contains more than thirty-two buffer definitions for a single pool, only the first thirty-two are used. Remaining definitions are ignored.

DFSSPMxx is not required to initialize IMS. It is intended only to override existing IMS storage pool definitions; therefore, no default member is supplied.

The format of the buffer definition override statement is as follows:



**FPL=**

Is the required keyword for IMS storage pool buffer override definitions. This keyword must be in position 1. The FPL= statement cannot be continued; however, more than one statement is allowed per pool.

**poolname**

Is the four-character name of the IMS storage pool. The valid names are:

AOIP  
 CESS  
 CIOP  
 EMHB  
 FPWP  
 HIOP  
 LUMC  
 LUMP

**size**

Specifies the buffer size, in bytes. The value must be a one- to five-digit

number between 8 and 65536. The buffer size is rounded up to the next multiple of 8. If the rounded value is not unique, it is discarded. You can specify values in 1K increments; that is, 1K, 2K, 3K, up to 64K.

#### **pbuf**

Specifies the number of buffers in the primary storage allocation. The value must be a one- to five-digit number between 2 and 65535. If the size of the primary allocation exceeds the upper expansion limit, the default is 2.

#### **sbuf**

Specifies the number of buffers in secondary storage allocations. The value must be a one- to five-digit number between 2 and 65535. If the size of the primary allocation exceeds the upper expansion limit, the default is 2.

#### **init**

Specifies whether (Y) or not (N) the primary storage allocation is obtained during IMS initialization. If the primary storage allocation is not obtained during initialization, it is not obtained until it is necessary to satisfy a buffer request. If an upper expansion limit is specified for the pool, and the size of either the primary or secondary storage allocation exceeds the upper expansion limit, the upper expansion limit is set to the default value of 2G-1.

If an FPL= is incorrectly specified, message DFS0639W is issued. One message is issued for each occurrence of an invalid parameter.

**Related Reading:** See *IMS Version 9: Messages and Codes, Volume 2* for more information on this message.

### **Examples of DFSSPMxx Statement**

The following are examples of how the DFSSPMxx PROCLIB member can be used to override the default buffer definitions for the CIOP pool.

```
FPL=CIOP, (248,20,15,Y), (500,20,10,N), (1016,15,8,Y)
FPL=CIOP, (2040,15,8,N), (4088,10,5,N), (8184,10,5,N)
FPL=CIOP, (16K,8,4,N), (32K,4,2,N)
```

The buffer sizes being defined range from 248 to 32768 (32K) bytes. Each buffer size is rounded up to the next multiple of 8 and then 8 bytes are added for internal processing. Table 44 shows the values used by the Storage Manager when allocating CIOP storage for the definition in the example.

*Table 44. Primary and Secondary Buffer Allocations*

<b>Buffer Size</b>	<b>Primary Buffer Allocation</b>	<b>Secondary Buffer Allocation</b>	<b>Obtain Primary Allocation?</b>
256	20	15	Y
512	20	10	N
1024	15	8	Y
2048	15	8	N
4096	10	5	N
8192	10	5	N
16392	8	4	N
32776	4	2	N

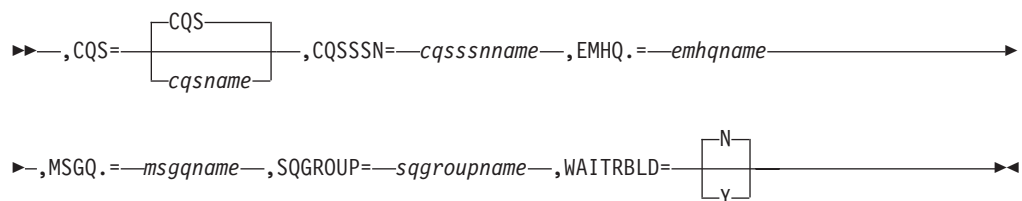
## Member DFSSQxxx

The DFSSQxxx PROCLIB member specifies parameters related to the shared message queues and the CQS address space. The DFSSQxxx PROCLIB member applies to the DB/DC and DCCTL environments. You can find an example of the DFSSQxxx PROCLIB member, DFSSQ999, in library SDFSSLIB.

### Control Statements

A DFSSQxxx member consists of one or more 80-character records. Keyword parameters are coded in positions 1-71 in free format and can contain leading and trailing blanks. Positions 72-80 are ignored. Multiple keywords can be specified in each record; commas are used to delimit keywords.

Errors encountered during DFSSQxxx PROCLIB member processing cause IMS initialization to terminate.



#### CQS=

Specifies the one- to eight-character name of the PROCLIB member that contains one of the following:

- The procedure for the CQS address space.

When shared message queues are requested the IMS control region automatically starts the CQS procedure during IMS initialization. The default procedure name is CQS.

- The START command.

The START command begins in column 1 with the characters START. The START command and its parameters must not extend beyond column 71.

When shared message queues are requested, the IMS control region issues the user-specified START command to start the CQS address space.

#### CQSSSN=

Specifies the one-to four- character subsystem name of the CQS address space. IMS uses this name to connect to the proper CQS address space. When connecting IMS to CQS, you must specify the same value on CQSSSN= and on the SSN= parameter of the CQSIPxxx PROCLIB member for the target CQS. The parameter is required and no default exists.

As many as 32 different IMS control regions can specify the same CQSSSN= parameter.

#### EMHQ=

Specifies the one- to sixteen-character name of the primary structure that contains the shared expedited message handler queues (EMHQs). In a shared-queues environment when Fast Path is installed, the presence of this statement indicates that the EMHQ structure, along with its associated log structure, checkpoint data set, and structure recovery data set, are required. To disable shared expedited message handler processing, you can remove the EMHQ statement from this member, and the EMHQ structure, along with its associated log structure, checkpoint, and structure recovery data sets, are not required.

If you do not want to utilize an EMHQ structure for an IMS system, you need to delete the STRUCTURE statement for the EMHQ from the CQSSLxxx PROCLIB member. If you do not want to utilize an EMHQ structure for all of the IMS systems in a sysplex, you also need to delete the STRUCTURE statement for the EMHQ from the CQSSGxxx PROCLIB member.

When you do not use an EMHQ structure, you should also do the following:

- In the CFRM policy, delete the STRUCTURE definition for the EMHQ structure.
- In the z/OS Logger Structure Definition policy for the CQS log, delete the LOGSTREAM definition for the EMHQ structure.
- If you are using an IMS automation program for any command that deals with an EMHQ structure, delete the command in the automation.

**Related Reading:**

- For additional information on disabling shared expedited message handler processing, see *IMS Version 9: Administration Guide: Transaction Manager*
- For additional information on the CQSSLxxx and CQSSGxxx PROCLIB members, see *IMS Version 9: Common Queue Server Guide and Reference*

If you specify an EMHQ= structure, you must also specify shared message queues on the MSGQ= parameter. IMS requires the message queues to process system messages that result from Fast Path activity. If the EMH queues are built in a coupling facility structure, the message queues must also be in a structure.

The name you specify for EMHQ= must not be the same as the name you specify for MSGQ=. The name must also be specified to CQS on the STRNAME= parameter in the CQSSLxxx and CQSSGxxx PROCLIB members.

**Related Reading:** For additional information on the structure name, see *IMS Version 9: Base Primitive Environment Guide and Reference* or *IMS Version 9: Common Queue Server Guide and Reference*.

This parameter is valid only when Fast Path is installed.

**MSGQ=**

Specifies the one- to sixteen-character name of the primary structure that contains the shared message queues. If the message queues are built in a coupling facility structure, the EMH queues, if used, must also be in a structure.

The name you specify for MSGQ= must not be the same as the name you specify for EMHQ=. The name must also be specified to CQS on the STRNAME= parameter in the CQSSLxxx and CQSSGxxx PROCLIB members.

**Related Reading:** For additional information on the structure name, see *IMS Version 9: Base Primitive Environment Guide and Reference* or *IMS Version 9: Common Queue Server Guide and Reference*.

**SQGROUP=**

Specifies a one- to five- character identifier. IMS concatenates this identifier to DFS to create the XCF IMS shared queues group name. You must specify the same identifiers for all IMS subsystems that share the same set of structures. You can use the same identifier for the CQSGROUP= parameter in the CQSIPxxx PROCLIB member. This parameter is required and there is no default.

**Related Reading:** For information on the CQSGROUP= parameter, see *IMS Version 9: Base Primitive Environment Guide and Reference* or *IMS Version 9: Common Queue Server Guide and Reference*.



**WAITRBLD=Y|N**

Specifies whether activity against the EMHQ structure waits while the structure is being rebuilt. If you specify Y (yes), all activity against the EMHQ structure waits until the structure rebuild completes. If you specify N (no), activity against the EMHQ structure continues while CQS rebuilds the structure.

The value specified for this keyword remains in effect for the life of the structure and cannot be changed.

**Restriction:** This parameter does not apply to the MSGQ. The MSGQ must wait for rebuild to complete before activity can resume.

If you change the WAITRBLD=parameter, you must first deallocate the fast path structure.

**Member DFSVSMxx**

The control statements in this member are processed during IMS initialization. The following types of control statements are allowed:

1. The subpool definition statements, VSRBF, IOBF, DEDB, DEDBMAS, and DBD, are used to define the size and number of buffers in a subpool.
2. The shared resource pool definition statements, RESVPOOL and POOLID, control the number and use of local shared resource pools.
3. The SBONLINE statement is used to specify sequential buffering in an online IMS or DBCTL system.
4. The OPTIONS, DLITRACE, and OSAMOP statements allow you to influence some of the performance and debugging facilities of IMS.
5. The OLDSDEF, WADSDEF, and ARCHDEF statements are used to define the initial DASD logging environment.
6. The NODYNALLOC statement is used to disable dynamic allocation for batch IMS.
7. The NOPDBO statement is used to disable parallel processing of full-function databases during database open.
8. The CFNAMES statement describes coupling facility structure names for sysplex data sharing.
9. The MADSIOT statement enables the long busy handling function. See *IMS Version 9: Administration Guide: System* for more information about MADSIOT and the long busy handling function.
10. The LOCKTIME statement is used to control the IRLM Lock Timeout function.

In an IMS batch system, these control statements are put in a data set with ddname DFSVSAMP. The data set must contain blocked or unblocked 80-character records.

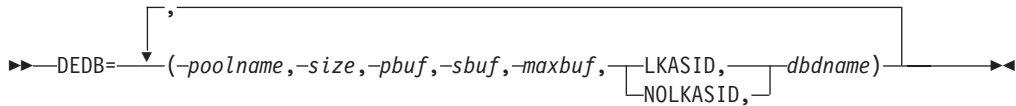
**Recommendation:** Do not place DFSVSAMP in the same PDS as your user application files. This causes batch jobs to abend with ABEND0C4 in DFSRTM00.

In an IMS online system, these control statements are put in the IMS.PROCLIB data set in member DFSVSMxx. The VSPEC symbolic parameter in the IMS, DBC, and DCC procedures is used to specify the suffix (xx) of the DFSVSMxx member to be used. During stage 2, IMS creates a default member with suffix 00. If a suffix is not specified on the startup procedure, the default member is used. The default values are also used if the control statements in the startup procedure are missing or invalid.

## Defining Fast Path DEDB Buffer Pools for Single-area Structures

This section describes control statements in the DFSVSMxx member that you use to define Fast Path DEDB buffer pools for single-area structures.

**Control statements:** The format of the control statements for defining a Fast Path buffer pool for a single-area structure is:



### DEDB=

Is the required keyword for defining Fast Path buffer pools. It must be in the first position of the control statement.

#### poolname

The one- to eight-character name for the pool. The poolname is used as identification on display terminals and reports.

#### size

A three- to five-digit number specifying the size of the pool. All of the standard DEDB supported buffer sizes are supported. You can express sizes numerically, or as standard CI sizes, 1K, 2K, 4K.

#### pbuf

The primary buffer allocation. Specify a value from 1 to 32766.

#### sbuf

The secondary buffer allocation. Specify a value from 1 to 9999. This secondary allocation is used when the primary allocation runs low.

#### maxbuf

The maximum number of buffers allowed for this pool. It is a combination of PBUF plus some iteration of SBUF. The maximum value allowed is 32767. If maxbuf is specified greater than 32767, it will default to 32767.

#### LKASID | NOLKASID

Indicates whether this pool is to be used as a local cache with buffer lookaside capability.

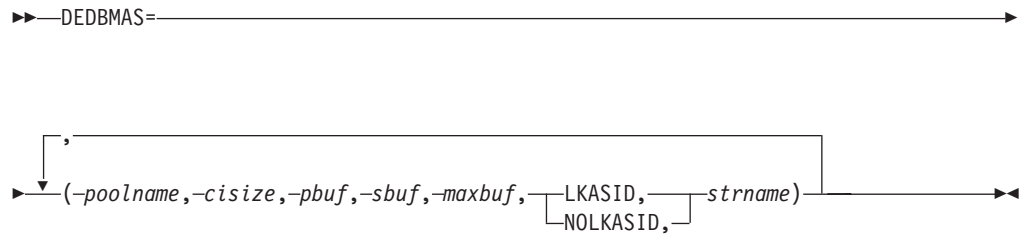
#### dbdname

Associates a pool to a specific area or DBD. If dbdname is an area name, then the pool is used only by that area. If dbdname is a DBD name, then the pool will be used by all the areas within that DBD.

## Defining Fast Path DEDB Buffer Pools for Multi-Area Structures

You can enable all DEDB areas that share a single coupling facility structure to use the same buffer pool. This section describes control statements in the DFSVSMxx member that you use to define Fast Path DEDB buffer pools for multi-area structures.

**Control statements:** The format of the control statements for defining a Fast Path buffer pool for a multi-area structure is:

**DEDBMAS=**

Defines a Fast Path buffer pool that is to be used by a multi-area structure.

**poolname**

The one- to eight-character name for the pool. The poolname is used as identification on display terminals and reports.

**csize**

The control interval size of the area. All areas that share a multi-area structure must have the same control interval size. If there is a discrepancy between the control interval size of the area used in creating the structure and the control interval size of the area attempting to share the structure, the open process for the area attempting to share the structure fails.

**pbuf**

The primary buffer allocation. Specify a value from 1 to 32766.

**sbuf**

The secondary buffer allocation. Specify a value from 1 to 9999. This secondary allocation is used when the primary allocation runs low.

**maxbuf**

The maximum number of buffers allowed for this pool. It is a combination of PBUF plus some iteration of SBUF. The maximum value allowed is 32767. If maxbuf is specified greater than 32767, it will default to 32767.

**LKASID | NOLKASID**

Indicates whether this pool is to be used as a local cache with buffer lookaside capability.

**strname**

The required one- to sixteen-character name of the primary coupling facility structure.

The installation must have defined the structure in the coupling facility resource management (CFRM) administrative policy. The structure name must follow the naming rules of the CFRM. If the name has fewer than 16 characters, the system pads the name with blanks. The valid characters are A-Z, 0-9, and the characters \$, &, # and \_. Names must be uppercase and start with an alphabetic character.

**Restriction:** Avoid using names that IBM uses for its structures. Do not begin structure names with the letters A-I, or the character string SYS.

**Related Reading:** For more information about shared VSO and multi-area structures, refer to *IMS Version 9: Administration Guide: Database Manager*.

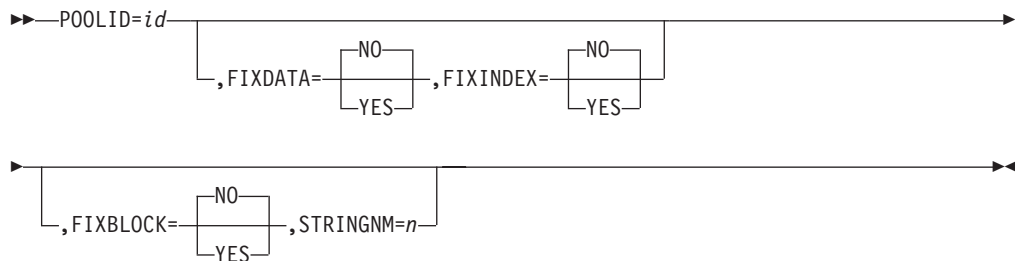
**Defining VSAM Buffer Pools**

This section describes control statements in the DFSVSMxx member that you use to define multiple VSAM local shared resource pools.

**Related Reading:** For overview information, refer to “Specifying IMS Buffer Pools” on page 430.

One or more VSAM local shared resource pools can be defined through the subpool definition statement POOLID. One POOLID statement is specified for each shared resource pool. The POOLID statement must be followed by one or more VSRBF subpool definition statements, which define the subpools within that shared pool.

**Control statements:**



**POOLID=**

Is the required keyword for VSAM shared resource pool definition. It must begin in the first position of the control statement. Only one set of subparameters can appear on each control statement. Each statement defines one shared resource pool. The POOLID statement is used with the DBD statement to direct a given data set to a specific shared resource pool.

Only the first VSAM local shared resource pool defined with the POOLID statement can contain VSAM subpool definition statements (VSRBF) for subpools not dedicated to specific data sets. All VSAM local shared resource pools defined after the first pool must be dedicated to specific data sets. This is done by adding a DBD statement that defines which data sets will use the shared resource pool. Therefore, defining a VSAM local shared resource pool other than the first pool defined, without a DBD statement referencing that pool will result in a failure for the POOLID statement.

POOLIDs 0 and 1-15 can be used. The total number of POOLIDS must not exceed 16. Also refer to the paragraph under *RESVPOOL=*.

**id** Is a one- to four-character alphanumeric field that specifies a user identifier assigned to a shared resource pool. This parameter is required. It is used with the DBD statement to direct a given data set to a specific shared pool.

**FIXDATA=YES|NO**

Is an optional parameter that specifies the data shared resource pool long-term-page-fixing option. If YES is specified, all buffers in the data shared resource pool are long-term-page-fixed at initialization of the shared resource pool. If NO is specified, no buffers in the data shared resource pool are long-term-page-fixed. If this parameter is omitted, the default is NO.

**FIXINDEX=YES|NO**

Is an optional parameter that specifies the index shared resource pool long-term-page-fixing option. If YES is specified, all buffers in the index shared resource pool are long-term-page-fixed at initialization of the shared resource pool. If NO is specified, no buffers in the index shared resource pool are long-term-page-fixed. If this parameter is omitted, the default is NO.

**FIXBLOCK=YES|NO**

Is an optional parameter that specifies the I/O-related control blocks long-term-page-fixing option. If YES is specified, all I/O-related control blocks are long-term-page-fixed at initialization of the shared resource pool. If NO is specified, no I/O-related control blocks are long-term-page-fixed. If this parameter is omitted, the default is NO.

**STRINGNM=n**

Is an optional parameter that specifies the maximum number of VSAM I/O requests that can be concurrently active. It can be used to override the MAXREGN parameter specified at system definition time and the PST value on the EXEC statement. The specified value must be a decimal number between 1 and 255. The default is 255. The value specified should be a number as close as possible to the maximum number of regions expected to be running concurrently, including those that could be dynamically started. If the value is omitted or invalid, the default is the MAXREGN value or the PST value, which, if present, overrides the MAXREGN value.

The buffer fix and string number parameters must be separated by commas and coded without intervening blanks. They can be coded in any order. If any parameter is invalid, the remainder of the statement is ignored and defaults apply for the remaining parameters.

The POOLID buffer fix and string number values override the VSAMFIX and STRINGMX parameters on the OPTIONS statement for that particular shared pool.



**RESVPOOL=shared-pool-number**

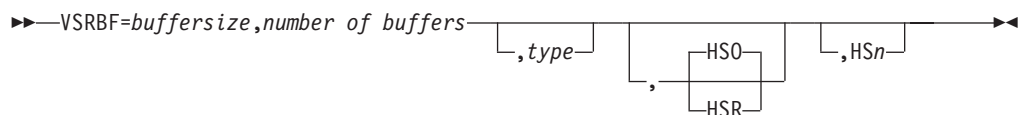
*shared-pool-number* represents the VSAM local shared resource pool number of the shared pool that IMS is not allowed to use. One or more numbers can be specified in any order and must be separated by commas, but without any blanks. The range is from 1 to 15. Shared resource pool 0 is reserved for IMS and is not allowed. A blank must follow the last number specified, and the remaining portion of the statement is not examined.

The RESVPOOL statement is only applicable if you have a product other than IMS using VSAM local shared resource pools that IMS is not allowed to use.

**Defining VSAM Subpools**

This section describes control statements in the DFSVSMxx member that you use to define VSAM subpools.

**Control statements:**



**VSRBF=**

The keyword for VSAM subpool definition. It must begin in the first position of the control statement. Only one set of subparameters can appear on each control statement. Each statement defines one subpool.

**buffersize**

A three- to five-digit number specifying the buffer size for this subpool. Acceptable values are 512, 1024, 2048, 4096, 8192, 12288, 16384, 20480, 24576, 28672, and 32768.

**number-of-buffers**

A one- to five-digit number (3 to 32767) specifying the number of buffers in this subpool. If you specify a number of buffers less than the minimum number required, IMS increases the number to the minimum and issues a warning message.

**type**

A one-character field that specifies whether the subpool in the shared resource pool is an index subpool (I) or a data subpool (D). This parameter is optional. If this parameter is invalid or not specified, data subpool (D) is assumed.

If you do not specify a value for type in any VSRBF statements, both data and index occupy the same buffer subpool. When you do specify D or I in any VSRBF statement, you must also specify a type of I for each size that might be required to handle index buffering requests. An error with the message DFS0730I RC 0,DC might result if a subpool of sufficient size is not found.

The following example provides buffers for data (2K) and index (4K) if these are the only VSRBF statements in DFSVSMxx.

```
VSRBF=2048,20
VSRBF=4096,40
```

In the next example, ten index buffers (2K) are provided in the first line, and ten data buffers (2K) are provided in the second line. The third line provides 20 data buffers (4K). No index buffers of 4K are provided in this example.

```
VSRBF=2048,10,I
VSRBF=2048,10,D
VSRBF=4096,20
```

Although the index and data components of a VSAM data set must reside in the same shared resource pool, they can be assigned different subpools. You can define subpools that accommodate each component of the data set. Note that an index subpool cannot exist without a data subpool (within a given shared resource pool), but a data subpool can exist without an index subpool. If any index subpool exists in a local shared resource pool, all index components of VSAM data sets in that shared pool must share the index subpool. The CONTROLINTERVALSIZE parameter in the listing of the VSAM catalog that shows the control interval sizes assigned to the components can help you determine the buffer sizes needed for both index and data components.

**HSO|HSR**

Optionally specifies the kind of action IMS should take if Hiperspace (extended storage on z/OS) buffering for this subpool is not available. If this parameter is invalid or not coded when HS<sub>n</sub> is given, the HSO option is assumed.

**HSO**

Indicates that Hiperspace buffering is optional and IMS can continue initialization without Hiperspace buffering.

**HSR**

Indicates that Hiperspace buffering is required and IMS must terminate if Hiperspace buffering is not available.

HSO and HSR are the minimum truncations of which up to the full terms (HSOptional and HSRequired) are recognized.

**HSn=**

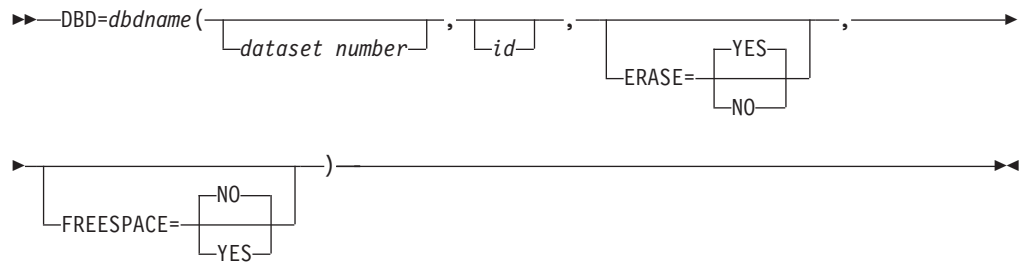
Is an optional one- to eight- digit number *n* ranging from (3 to 16777215) that specifies the number of Hiperspace buffers to build for this subpool.

HSO|HSR and HSn are valid only on 4K and larger boundaries. See the following example.

```
4096,10,HS40,HSR
8192,10,HS50,HSRE
12288,4,HS12
16384,4,HS20,HSO
20480,4,HSOP,HS8
24576,4,HSOPTIONAL,HS8
28672,1,HS9,HSREQUIRED
32768,1,HSREQ,HS9
```

Only the buffer size and number-of-buffers parameters are positional. The type and HS parameters can be coded in any order but must be separated by commas without any intervening blanks. If any parameter is invalid, the remainder of the statement is ignored and defaults apply for the remaining parameters.

Using the VSRBF statement without any POOLID statement can still define one, and only one, VSAM local shared resource pool. However, even with one VSAM shared resource pool, you can now use the TYPE parameter to define either an index or data subpool within the shared resource pool.



**DBD=**

If coded, specifies that the data set having a matching ID parameter, as defined on the POOLID shared pool definition statement, is to be directed to the indicated shared resource pool.

A shared resource pool is assigned to a data set based upon the shared resource pool identifier specified on the DBD statement for that data set. Then a subpool within the assigned shared resource pool is assigned to the data set based upon buffer length.

If no subpool in the shared resource pool is big enough, an attempt to open the data set fails. An open is then retried in the default shared pool, which is the first shared resource pool you define. To avoid another open failure, subpools of the largest sizes that might be encountered in the system should be defined in the default pool.

If no DBD statement is coded for a data set, the default shared resource pool is assigned.

**dbdname**

Specifies the name on the DBD macro statement NAME= keyword. The DBD name can be for a non-partitioned database, a HALDB partition, or a HALDB master. For a HALDB partition, specify the name of the partition as the dbdname. If you specify a HALDB master name, all of the HALDB partitions associated with the HALDB are assigned to the same shared resource pool.

**Related Reading:** See “Database Description (DBD) Generation” in *IMS Version 9: Utilities Reference: Database and Transaction Manager* for information on coding the DBD macro statement.

**data set number**

Identifies the specific data set of a data set group within a database (identified by the dbdname parameter) that requests assignment of a specific shared pool. The number is an IMS internally assigned value between 1 and 10.

For data organizations such as primary index, unique secondary index, and HISAM without dependent segments, the primary data set of the data set group is assigned data set number 1. No secondary data set of the data set group exists for these data organizations.

For data organizations such as non-unique secondary index and HISAM with dependent segments, the primary data set of the data set group is assigned data set number 1, and the secondary data set of the data set group is assigned data set number 2.

For hierarchic direct data organization, the data set group always consists of a single data set. The data set of the first data set group is assigned data set number 1, and data set numbers for subsequent data set groups are sequentially assigned numbers 2,3,...10. The maximum number of data set groups for a database is 10.

**Requirement:** For High Availability Large Databases (HALDBs), specify the data set number as an alphabetic character. The valid data set numbers defined for HALDB partition data sets are A through J, L, and X. (When HALDB online reorganization is used, data sets M through V and Y are automatically directed to the same shared resource pools for data sets A through J and X. The specification of M through V and Y are not valid.) The data set of the first data set group must be assigned the letter A (for both data sets A and M when HALDB online reorganization is used). The data set numbers for subsequent data set groups (B through J and N through V when HALDB online reorganization is used) must be sequentially assigned the other valid letters (B,C, ..., J). Use the letter L to specify the indirect list data set. For PHIDAM databases, use the letter X (for both data sets X and Y when HALDB online reorganization is used) to specify the primary index data set.

For PSINDEX databases, you cannot code an alphanumeric to designate the data set number. You must code a numeric value.

**id** Specifies the user-defined identifier that is assigned to a specific shared resource pool. The ID is a one- to four-character alphanumeric field and must be equal to the ID assigned to the specific shared resource pool on the POOLID statement.



**ERASE=YES|NO**

Indicates the treatment of logical records that are deleted. YES indicates that the deleted record should be erased. NO indicates that the deleted record should not be erased but should be marked as a deleted record. The default is YES.

**FREESPACE=YES|NO**

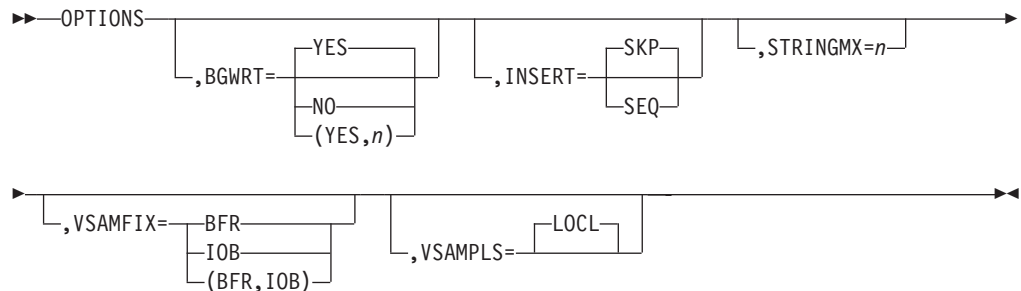
Indicates the treatment of the defined free space percent in the KSDS. If YES is specified, the defined free space should be preserved. If NO is specified, the defined free space should not be preserved. The default is NO.

**Defining VSAM Performance Options**

This section describes the OPTIONS control statement parameters in the DFSVSMxx member that can be used to influence the performance of the IMS DL/I buffering services.

The word OPTIONS starting in position one identifies the OPTIONS statement. You can specify the parameters in any order, but you must separate them with commas. A blank must follow the last parameter. The remaining portion of the statement is not examined. You cannot continue an OPTIONS statement on a subsequent line, but you can provide several statements. If an OPTIONS parameter appears more than once, its setting is determined by the last occurrence. The OPTIONS statement and all of its parameters are optional.

**Control Statements:**



**BGWRT=**

Specifies whether (YES) or not (NO) the background write (BGWRT) function of the buffer handler is to be active. You can also activate background write by coding BGWRT=(YES,*n*) or omitting the parameter. *n*: is a two-digit number from 10 to 99 specifying the percentage of buffers in each subpool to be considered as candidates for writing by the background write function; the default is 34.

|  
|  
|

**Related Reading:** For an explanation of background write, see “Determining Which VSAM Options to Use” in *IMS Version 9: Administration Guide: Database Manager*.

**BLDSNDX=**

Provides an option to control the building of secondary indexes during initial load of an HALDB.

If BLDSNDX=YES, or the parameter is omitted, then the creation of secondary indexes during initial load is allowed.

IF BLDSNDX=NO, then the creation of secondary indexes during initial load is suppressed.

**INSERT=**

Specifies the insert mode that the buffer handler uses when inserting new KSDS logical records into any of the following databases: HISAM, Single segment HISAM, and INDEX.

If a program inserts many new root segments in sequence by key, specifying INSERT=SEQ causes the buffer handler to use VSAM sequential mode PUTs. VSAM leaves free space (if you specify it in the DEFINE) for VSAM data sets in CIs created for the new records that are inserted using VSAM's mass insert. Specifying INSERT=SKP, or omitting the parameter, causes the buffer handler to use VSAM skip sequential mode PUTs if the database is in sequential mode, or to use direct mode PUTs if the database is not in sequential mode. For more information on sequential mode databases, see "VSAM Subpool Definition" on page 430. In either mode, VSAM does not leave free space, even if it is specified in the DEFINE regardless of whether DL/I inserts in load or insert mode.

The default is SKP, except in utilities, where it is SEQ.

You can override INSERT=SEQ with the FREESPACE parameter of the DBD control statement. FREESPACE=YES causes the buffer handler to use VSAM sequential mode PUTs. Specifying FREESPACE=NO, or omitting the parameter, causes the buffer handler to use VSAM skip sequential mode PUTs if the database is in sequential mode, or to use direct mode PUTs if the database is not in sequential mode.

**STRINGMX=**

Specifies the maximum number of VSAM requests that can be concurrently active. *n* is any number between 1 and 255. If you omit this keyword, the default value is the value specified on either the MAXREGN= parameter on the IMSCTRL macro or the PST= parameter on the IMS procedure EXEC statement (which overrides the MAXREGN= value). The STRINGMX= keyword overrides either of these two values.

**VSAMFIX=**

Causes VSAM resource pools to be page fixed in main storage.

**BFR**

Specifies that the buffer subpools and control blocks should be fixed. If you define a large number of buffers and then use VSAMFIX=BFR the system might abend.

**IOB**

Specifies that the input/output related blocks should be fixed.

Either or both operands can be specified. The syntax for specifying both operands is (BFR,IOB) or (IOB,BFR).

**VSAMPLS=LOCL**

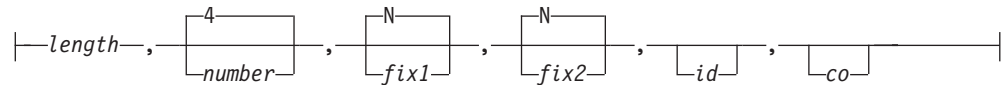
Specifies that the VSAM shared resource pools are to be built. The only valid value is the default, LOCL.

**Defining OSAM Subpools**

This section describes control statements in the DFSVSMxx member that you use to define OSAM subpools.

**Control statements:**

▶▶ IOBF=(— AA —) ▶▶

**AA:****IOBF=**

Is the required keyword for OSAM subpool definition. It must begin in the first position of the control statement. Only one set of subparameters can appear on each control statement.

**length**

Specifies the length of the buffers in the subpool. Specify the value in bytes, between 512 and 32000. Depending on the value that you specify, IMS rounds it up to 512, 1024, 2048, and thereafter to multiples of 2048. You can code specifications of 1024 and above as 1K, 2K, 4K, and thereafter round them up to multiples of 2K to a maximum of 32K. If you enter an invalid value, IMS ignores the entire IOBF statement. This parameter is required.

**number**

Specifies the number of buffers in the subpool. If specified, you must choose a value between 4 and 32767. If you choose a number greater than 32767, the default value of 255 is used. If not specified, the default value is 4. If this parameter is invalid, the remainder of the entry is ignored, and defaults apply for all remaining parameters. This parameter is optional.

**fix1**

Specifies the buffer long-term page-fixing option. If you specify Y, all buffers and buffer prefixes associated with this subpool are long-term page-fixed at initialization of the subpool. If you specify N, no buffers associated with this subpool are long-term page-fixed at initialization of the subpool. The default is N. This parameter is optional.

**fix2**

Specifies the buffer prefix long-term-page-fixing option. If you specify Y, all buffer prefixes associated with this subpool and the subpool header are long-term-page-fixed at initialization of the subpool. If you specify N, the subpool header and all buffer prefixes associated with this subpool are not long-term-page-fixed at initialization of the subpool. The default is N. This parameter is optional.

**id** Specifies a user-defined identifier to be assigned to a subpool. This ID is a one- to four-character alphanumeric field and is used with the DBD statement to assign a specific subpool to a given data set. If this parameter is not coded, IMS assigns a null ID to the subpool.

If two or more subpool definition statements specify the same buffer size (without the subpool ID), the number of buffers from the statements are summed. If the total does not exceed 32767, IMS builds a single subpool with the total number of buffers; however, if it does exceed 32767, a single subpool with 255 buffers is built.

**co** Specifies the subpool caching option. For more information on OSAM data caching, see

**Related Reading:** You can specify the caching option in one of the following ways: see "OSAM Subpool Definition" on page 431.

**omitted** No data caching.

	Caching is not active for the subpool.
<b>N</b>	No data caching.
	Caching is not active for the subpool.
<b>A</b>	Cache all data.
	Write all data read from DASD and all changed data to the coupling facility
<b>C</b>	Cache only changed data.
	Write all changed data written to DASD to the coupling facility.

If you specify any value other than A, C, or N, no data caching for the subpool occurs. The default is N.

### Control Statements:

▶▶ `DBD=dbdname(data set number,id)` ▶▶

### DBD=

If coded, specifies that the indicated subpool is to be assigned to the data set having a matching ID parameter as defined on the IOBF= subpool definition statement. For OSAM, ERASE and FREESPACE are invalid.

#### dbdname

Specifies the name on the DBD macro statement NAME= keyword. The DBD name can be for a non-partitioned database, a HALDB partition, or a HALDB master. For a HALDB partition, specify the name of the partition as the dbdname. If you specify a HALDB master name, all of the HALDB partitions associated with the HALDB are assigned to the same subpool.

**Related Reading:** For information about coding the DBD macro statement, see *IMS Version 9: Utilities Reference: Database and Transaction Manager*, "Database Description (DBD) Generation."

#### data set number

Identifies the specific data set of a data set group within a database (identified by the dbdname parameter) that requests assignment of a specific shared pool. The number is an IMS internally assigned value between 1 and 10.

For data organizations such as primary index, unique secondary index, and HISAM without dependent segments, the primary data set of the data set group is assigned data set number 1. No secondary data set of the data set group exists for these data organizations.

For data organizations such as non-unique secondary index, and HISAM with dependent segments, the primary data set of the data set group is assigned data set number 1, and the secondary data set of the data set group is assigned data set number 2.

For hierarchic direct data organization, the data set group always consists of a single data set. The data set of the first data set group is assigned data set number 1, and data set numbers for subsequent data set groups are sequentially assigned numbers 2,3, ...10. The maximum number of data set groups for a database is 10.

|  
|  
|  
|  
|  
|  
|  
|  
|  
|  
|  
|  
|  
|  
|

**Requirement:** For High Availability Large Databases (HALDBs), specify the data set number as an alphabetic character. The valid data set numbers defined for HALDB partition data sets are A through J, L, and X. (When HALDB online reorganization is used, data sets M through V and Y are automatically directed to the same shared resource pools for data sets A through J and X. The specification of M through V and Y are not valid.) The data set of the first data set group must be assigned the letter A (for both data sets A and M when HALDB online reorganization is used). The data set numbers for subsequent data set groups (B through J and N through V when HALDB online reorganization is used) must be sequentially assigned the other valid letters (B,C, ..., J). Use the letter L to specify the indirect list data set. For PHIDAM databases, use the letter X (for both data sets X and Y when HALDB online reorganization is used) to specify the primary index data set.

**id** Specifies the user-defined identifier to be assigned to a specific subpool. The ID is a one- to four-character alphanumeric field and must be equal to the ID assigned to the specific subpool on the IOBF statement.

Subpools are assigned to data sets based upon buffer length. First, a subpool having buffers equal to or greater than the buffer length required for the data set is located. Then, if a specific subpool was requested, it is assigned, if its length is not less than the required length. In this case, the first subpool that meets the length criterion is assigned. The subpool might not have an ID assigned to it.

▶▶—OSAMOP—IOSCB=NO—◀◀

The OSAMOP statement allows options specific to the OSAM access method.

**IOSCB=NO**

Indicates that no OSAM resources are to be allocated to the batch applications that do not require OSAM services.

▶▶—OSAMOP—OSAMGTF=YES—◀◀

**OSAMGTF=YES**

Enables OSAM to generate GTF trace records while in batch.

**Specifying Sequential Buffering for an Online System**

|  
|  
|  
|

Use the SBONLINE control statement in the DFSVSMxx member to request sequential buffering for an IMS or CICS system. This statement applies to the IMS DB/DC or DBCTL environments. SBONLINE requests loading of SB modules and specifies the maximum amount of buffer space that SB can use.

The format of the SBONLINE control statement is as follows:

**Control statements:**

▶▶—SBONLINE—,MAXSB=nnnn—◀◀

**SBONLINE =**

Specifies that sequential buffering modules are to be loaded in the IMS DB/DC or DBCTL environments, thereby allowing subsequent use of sequential buffering.

By default, IMS does not load the SB modules. This prevents an increase in virtual storage requirements.

**MAXSB=nnnn**

Specifies the maximum amount of space (in kilobytes) that can be allocated to SB buffers. This number represents the total amount of space to be allocated for **all** concurrently executing programs, not for just one individual program. Setting a limit on the amount of SB space protects you from allocating excessive virtual storage. The default for MAXSB is unlimited space.

When the MAXSB limit is reached, IMS stops allocating SB buffers until terminating programs release buffer storage.

**Defining Serviceability and Trace Options**

This section describes the OPTIONS control statement parameters in the DFSVSMxx member, which can be used to influence the serviceability of IMS. For example, the OPTIONS statement allows you to specify which IMS traces should be activated automatically by the system during IMS initialization.

The word OPTIONS starting in position 1 identifies the OPTIONS statement. You can specify the parameters in any order, but you must separate them with commas. A blank must follow the last parameter. The remaining portion of the statement is not examined. You cannot continue an OPTIONS statement on a subsequent statement, but you can provide several statements. If an OPTIONS parameter appears more than once, its setting is determined by the last occurrence. The OPTIONS statement and all of its parameters are optional.

**Recommendation:** Select ON for the following parameters:

- DL/I=
- LOCK=

For an online system, select ON for these additional parameters:

- DISP=
- SCHED=

Activating the trace tables specified by these parameters does not cause a noticeable performance impact and each of the trace tables can be extremely helpful to you in diagnosing a variety of problems that might occur in your environment. For certain types of problems, the IMS Support Center needs the trace output. If the trace output is not available, you might need to re-create the problem, which will take time and delay the resolution of the problem.

**Related Reading:** For more information about all of these traces, see *IMS Version 9: Diagnosis Guide and Reference*.

**Valid Environments:** The following is a list of serviceability and trace options that can be activated for particular environments. If you attempt to activate a trace in an environment that does not support that type of trace, IMS will ignore your request.

**Batch** The valid serviceability and trace options for a batch environment are:

- DL/I=
- DLOG=

- DUMP=
- DUMPIO=
- ISSUE681=
- ISSUE840=
- LOCK=

**DB/DC**

All trace types, except for LRTT=, are valid in a DB/DC environment.

**DBCTL**

The valid serviceability and trace options for a DBCTL environment are:

- CSLT
- DISP=
- DL/I=
- DLOG=
- DUMP=
- DUMPIO=
- FPTT=
- ISSUE681=
- ISSUE840=
- LATC=
- LOCK=
- OCMD
- ORTT=
- QMGR=
- SCHD=
- SQTT=
- STRG=
- SUBS=

**DCCTL**

The valid serviceability and trace options for a DCCTL environment are:

- DISP=
- DL/I=
- DLOG=
- DUMP=
- DUMPIO=
- FAST=
- FPTT=
- IDC0=
- ISSUE681=
- ISSUE840=
- LATC=
- LUMI
- MSCT=
- OTMT=
- QMGR=
- RRST=

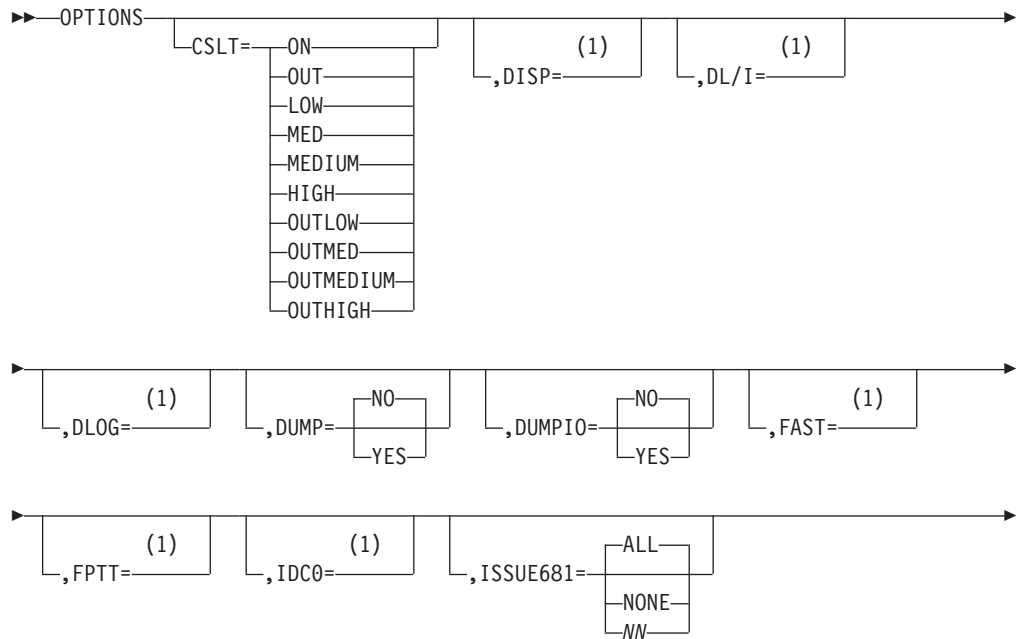
- SCHD=
- SQTT=
- STRG=
- SUBS=

**RSR Tracker**

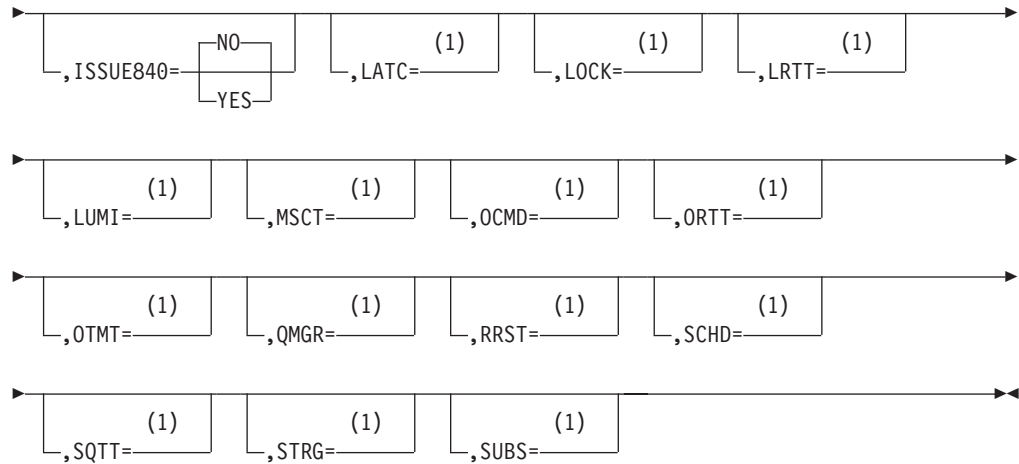
The valid serviceability and trace options for an RSR Tracker environment are:

- DISP=
- DL/I=
- DLOG=
- DUMP=
- DUMPIO=
- FAST=
- FPTT=
- IDC0=
- ISSUE681=
- ISSUE840=
- LATC=
- LOCK=
- LRTT=
- QMGR=
- SQTT=
- STRG=

**Control statements:**







**Notes:**

1 This parameter accepts values from the list of common options.

The following parameters accept a common set of options:

CSLT=	LATC=	QMGR=
DISP=	LOCK=	RRST=
DL/I=	LRTT=	SCHD=
DLOG=	LUMI=	SQTT=
FAST=	MSCT=	STRG=
FPTT=	OCMD=	SUBS=
IDC0=	OTMT=	

*Common Options:*

- ON** Turns on the trace table.
- OUT** Initializes and turns on the trace table for output to the log. Log records 4096 bytes long are written to the log when trace records are written.
- LOW** Turns on the queue manager trace for all low volume events.
- MED** Turns on the queue manager trace for all low and medium volume events.
- MEDIUM** Turns on the queue manager trace for all low and medium volume events.
- HIGH** Turns on the queue manager trace for all low, medium, and high volume events.
- OUTLOW** Initializes the trace table for output to the log and turns it on for tracing all low volume events.
- OUTMED** Initializes the trace table for output to the log and turns it on for tracing all low and medium volume events.
- OUTMEDIUM** Initializes the trace table for output to the log and turns it on for tracing all low and medium volume events.
- OUTHIGH** Initializes the trace table for output to the log and turns it on for tracing all low, medium, and high volume events.

**Related Reading:**

- See *IMS Version 9: Diagnosis Guide and Reference* for information about when and why trace tables are used.
- See *IMS Version 9: Operations Guide* for information about defining and setting up trace facilities.

**CSLT=**

Activates an IMS trace that traces activity related to IMS's interaction with the Common Service Layer. This includes IMS's interaction with OM, RM, and SCI. No default value exists. If this parameter is omitted, the trace is not activated by the OPTIONS statement. It can be activated and deactivated by the online /TRACE command.

For a list of the valid options for this keyword, see "Common Options" on page 399.

**DISP=**

Activates an IMS trace and traces the calls of the IMS dispatcher. No default value exists. If this parameter is omitted, the trace table is not activated by the OPTIONS statement. It can be activated and deactivated by using the online /TRACE command.

**Restriction:** DISP= is applicable to the online system only. The DISP trace is not allowed in a batch environment. If this trace is requested in a batch environment, the request will be ignored.

For a list of the valid options for this keyword, see "Common Options" on page 399.

**DL/I=**

Activates an IMS trace that traces the calls of certain DL/I modules and the functions invoked by these DL/I modules. The default value is ON. To deactivate the DL/I trace table, issue the /TRACE command in an online environment.

The DL/I trace, the program isolation trace, and the lock trace share the same trace table.

This keyword can also be used to turn on the DL/I retrieve trace table, but it does not control the number of table entries, which is fixed at 255.

For a list of the valid options for this keyword, see "Common Options" on page 399.

**DLOG=**

Activates an IMS trace. The DASD log trace traces the activity of the IMS DASD logger. No default value exists. If this parameter is omitted, the trace table is not activated by the OPTIONS statement. It can be activated or deactivated by using the /TRACE command.

For a list of the valid options for this keyword, see "Common Options" on page 399.

**DUMP=**

Provides a serviceability aid. If you specify NO or omit the parameter and an abnormal condition is encountered, the buffer handler issues a pseudoabend. If you specify YES and an abnormal condition is encountered, the buffer handler issues a standard abend, which results in the control region abnormally terminating also. If DUMP=YES and any abnormal termination occurs, the modified or newly created buffers in the pool at that time are not written to the data set.

**DUMPIO=**

Provides a serviceability aid for analyzing OSAM I/O errors. If you specify YES and an OSAM I/O error occurs, the IMS region abnormally terminates with a U0764. If the DUMPIO option is requested, the DFS07621 error message is not displayed.

**FAST=**

Activates an IMS trace that traces DBF trace entries from various Fast Path modules. No default exists. If this parameter is omitted, the trace is not activated by the OPTIONS statement. It can be activated and deactivated by using the online /TRACE command.

**Recommendation:** Run the FPTRACE in a test environment only. FPTRACE output is very large and can impact performance.

**Restriction:** The FAST trace is not allowed in a batch environment. If this trace is requested in a batch environment, the request will be ignored.

**Related Reading:** For more details on the FPTRACE, see *IMS Version 9: Command Reference*.

For a list of the valid options for this keyword, see “Common Options” on page 399.

**FPTT=**

Activates a Fast Path trace that traces certain Fast Path modules and the functions invoked by these Fast Path modules. The default value is OFF. To activate the Fast Path trace table, issue the /TRACE command in an online environment.

The Fast Path trace resides in its own trace table.

For a list of the valid options for this keyword, see “Common Options” on page 399.

**Related Reading:** For more details on the Fast Path trace, see *IMS Version 9: Command Reference*.

**IDC0=**

Activates the tracing of errors in modules DFSCNXA0 and DFSIDC00. No default value exists. If this parameter is omitted, the trace table is not activated by the OPTIONS statement. It can be activated or deactivated by using the online /TRACE command.

**Restriction:** The IDC0 trace is not allowed in a batch or DBCTL environment. If this trace is requested in either a batch or DBCTL environment, the request will be ignored.

For a list of the valid options for this keyword, see “Common Options” on page 399.

**ISSUE681=**

Specifies the number of DFS681I messages to be issued per second within a batch or BMP region.

**ALL**

Allows all DFS681I messages generated to be issued. All is the default.

**NONE**

Allows no DFS681I messages to be issued.

**ISSUE840=**

Specifies whether or not message DFS0848I will be issued if there is a duplicate segment in a unique secondary index, even if it is possible to back out prior changes for the call.

**NO** Specifies that DFS0848I message will not be issued for this condition. NO is the default.

**YES** Specifies that DFS0848I message will be issued for this condition.

**NN**

Is the number (1 - 99) of DFS681I messages that you allow to be issued. If some DFS681I messages are not issued, DFS683I is issued and states the number of DFS681I messages omitted.

**LATC=**

Activates an IMS trace. LATCH traces IMS latch activity. No default value exists. If this parameter is omitted, the trace table is not activated by the OPTIONS statement. It can be activated and deactivated by using by the /TRACE command.

**Restriction:** The LATC trace is not allowed in a batch environment. If this trace is requested in a batch environment, the request will be ignored.

For a list of the valid options for this keyword, see "Common Options" on page 399.

**LOCK=**

Activates an IMS trace that traces the lock activity of certain DL/I modules. The default value is ON. To deactivate the LOCK trace table, issue the /TRACE command in an online environment.

The lock trace, the program isolation trace, and the DL/I trace share the same trace table.

**Restriction:** The LOCK trace is not allowed in a DCCTL environment. If this trace is requested in a DCCTL environment, the request will be ignored.

For a list of the valid options for this keyword, see "Common Options" on page 399.

**LRTT=**

Activates an IMS trace that traces the activity of the log router component of the RSR tracking system. No default exists. If this parameter is omitted, the trace is not activated by the OPTIONS statement. It can be activated and deactivated by using the online /TRACE command.

**Restriction:** The LRTT trace is **only** allowed in an RSR Tracker environment. If this trace is requested in any other environment, the request will be ignored.

For a list of the valid options for this keyword, see "Common Options" on page 399.

**LUMI=**

Activates an IMS trace that traces LUM code that supports an LU 6.2 device. It contains the following information:

- Record written on entry and exit to selected LU 6.2 device modules
- Information from APPC/MVS verb with acceptable or unacceptable return code

If this parameter is omitted, the trace table is not activated by the OPTIONS statement. It can be activated and deactivated by using the online /TRACE command.

**Restriction:** The LUMI trace is not allowed in a batch, RSR Tracker, or DBCTL environment. If this trace is requested in one of these environments, the request will be ignored.

For a list of the valid options for this keyword, see “Common Options” on page 399.

**MSCT=**

Activates an IMS trace for Multiple Systems Coupling (MSC).

For a list of the valid options for this keyword, see “Common Options” on page 399.

**OCMD=**

Activates an IMS trace that traces IMSplex command activity. No default value exists. If this parameter is omitted, the trace is not activated by the OPTIONS statement. It can be activated and deactivated by the online /TRACE command.

For a list of the valid options for this keyword, see “Common Options” on page 399.

**ORTT=**

Activates an IMS trace that traces the Online Recovery System (ORS) activity in the IMS control region. If this parameter is omitted, the ORS trace is not activated by the OPTIONS statement. It can be activated and deactivated by using the online /TRACE command.

**Restriction:** The ORTT trace is not allowed in a batch or DC environment. If this trace is requested in this environment, the request is ignored.

**OTMT=**

Activates an IMS trace. OTMT traces the flow of control through IMS OTMA. No default exists. If this parameter is omitted, the trace is not activated by the OPTIONS statement. It can be activated and deactivated by using the online /TRACE command.

**Restriction:** The OTMT trace is not allowed in a batch, RSR Tracker, or DBCTL environment. If this trace is requested in any of these environments, the request will be ignored.

For a list of the valid options for this keyword, see “Common Options” on page 399.

**QMGR=**

Activates an IMS trace for an online system. QMGR traces calls made to the IMS queue manager. No default value exists. If you omit this parameter, the trace table is not activated by the OPTIONS statement. It can be activated and deactivated by using the online /TRACE command.

**Restriction:** The QMGR trace is not allowed in a batch environment. If this trace is requested in a batch environment, the request will be ignored.

For a list of the valid options for this keyword, see “Common Options” on page 399.

**RRST=**

Activates an IMS trace for Resource Recovery Service (RRS). No default value exists. If this parameter is omitted, the trace table is not activated by the OPTIONS statement. The trace can be activated and deactivated by using the online /TRACE command.

**Restriction:** The RRST trace is not allowed in a batch environment. If this trace is requested in a batch environment, the request will be ignored.

For a list of the valid options for this keyword, see “Common Options” on page 399.

**SCHD=**

Activates an IMS trace. SCHD traces the calls of the IMS scheduler. No default value exists. If this parameter is omitted, the trace table is not activated by the OPTIONS statement. It can be activated and deactivated by using the online /TRACE command.

**Restriction:** The SCHD trace is not allowed in batch or RSR Tracker environments. If this trace is requested in either of these environments, the request will be ignored.

For a list of the valid options for this keyword, see “Common Options” on page 399.

**SQTT=**

Activates an IMS trace for an online system. SQTT traces the shared queues interface including:

- Initialization of the connection to CQS
- CQS requests
- Notification of work available from CQS
- Termination of the connection to CQS

No default value exists. If you omit this parameter, the trace table is not activated by the OPTIONS statement, but can be activated and deactivated by using means of the online /TRACE command.

**Restriction:** The SQTT trace is not allowed in a batch environment. If this trace is requested in a batch environment, the request will be ignored.

For a list of the valid options for this keyword, see “Common Options” on page 399.

**STRG=**

Activates an IMS trace. STRG traces all calls to the IMS storage manager that require modifications to one of the following pools:

CESS  
CIOP  
EMHB  
FPWP  
HIOP  
LUMC  
LUMP

If this parameter is omitted, the trace table is not activated by the OPTIONS statement; instead, it can be activated and deactivated by using the online /TRACE command.

STRG= is applicable only to the online system.

**Restriction:** The STRG trace is not allowed in a batch environment. If this trace is requested in a batch environment, the request will be ignored.

For a list of the valid options for this keyword, see “Common Options” on page 399.

**SUBS=**

Activates an IMS trace. SUBS traces the connection and disconnection of DB2 subsystems to the IMS control region. No default value exists. If this parameter is omitted, the trace is not activated by the OPTIONS statement. It can be activated and deactivated by using the online /TRACE command.

**Restriction:** The SUBS trace is not allowed in batch or RSR Tracker environments. If this trace is requested in either of these environments, the request will be ignored.

For a list of the valid options for this keyword, see “Common Options” on page 399.

Table 45 is a summary of the traces that can be activated using the OPTIONS parameter. It shows the number of trace entries and the storage requirements for the various traces.

Table 45. Trace Entries and Storage Requirements

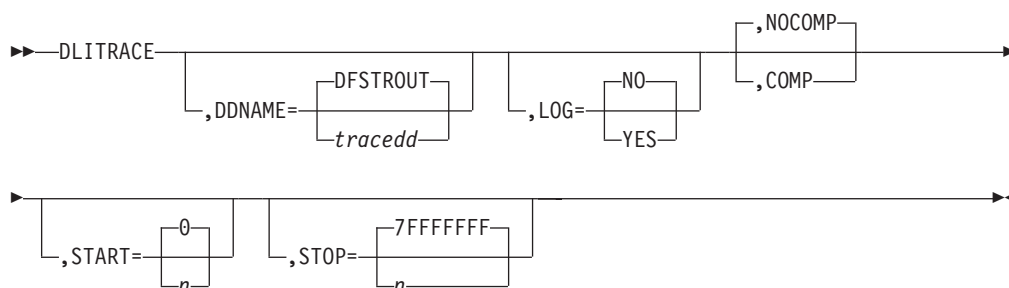
Type	No. of Trace Tables	Entry Size	Entries Per Table	Storage Bytes Required
Common Service Layer	8	20 Hex	126	32K
Dispatcher	10	20 Hex	126	40K
DLI/Lock	18	20 Hex	126	72K
DLog	6	20 Hex	126	24K
Fast Path	2	20 Hex	126	8K
Intercommunications	18	20 Hex	126	72K
Latch	12	20 Hex	126	48K
Log Router	8	X '20'	126	32K
LU 6.2	8	X '20'	126	32K
Multiple Systems Coupling	10	X '20'	126	40K
OCMD	8	X '20'	126	32K
Online Recovery	8	X '20'	126	32K
OTMA	8	X '20'	126	32K
Queue Manager	8	X '20'	126	32K
Resource Recovery Services	12	X '20'	126	48K
Scheduler	5	X '20'	252	20K
Shared Queues	8	X '20'	126	32K
Storage Manager	10	X '20'	126	40K
SubSystem	8	X '40'	63	32K

**Note:** Only DLog storage is below the 16-megabyte line. For all other types listed here, storage is above the 16-megabyte line.

**Related Reading:** For additional information on the traces that you can define using these keywords, see *IMS Version 9: Utilities Reference: System*.

### Defining DL/I Call Image Trace

This section describes control statements in the DFSVSMxx member that you use to specify a DL/I call image trace.

**Control statements:**

The DLITRACE statement invokes the tracing of DL/I calls in a batch environment. Specify the word DLITRACE in the first position to identify this statement. You can specify the parameters in any sequence, but you must separate them with commas; each parameter can be specified only once. If you don't specify any parameters, IMS uses the default values.

A DLITRACE statement cannot be continued on a subsequent statement. The first statement encountered in the DFSVSAMP data set establishes the options. Subsequent statements are bypassed, and message DFS2471 is issued. In cases where a parameter is repeated, the options remain set as specified by the first occurrence, and message DFS2471 is issued.

**Related Reading:** For details on the use of DL/I trace, see *IMS Version 9: Diagnosis Guide and Reference*.

**DDNAME=**

Indicates that sequential data set output is requested and gives the DD statement name (tracedd) to be present in the JCL. If you omit this parameter and do not supply output DD, DDNAME=DFSTROUT is assumed, and an attempt is made to route output to the sequential data set defined by DFSTROUT DD. However, if you omit this parameter and request log output, no attempt is made to open such a sequential data set.

**LOG=**

Specifies whether (YES) or not (NO) DL/I call image capture output is to be routed to the IMS system log. The default is NO, and a sequential output data set will contain the trace output.

**NOCOMP | COMP**

Specifies whether (COMP) or not (NOCOMP) DL/I Test Program COMPARE statements are generated for both PCB comparisons and data comparisons. The default is NOCOMP.

**Related Reading:** For more information, see *IMS Version 9: Application Programming: Database Manager*.

**START=**

Specifies a one- to eight-character hexadecimal value representing a count of DL/I calls issued by an application program against a specific PSB. This value indicates at which point in the program processing tracing should begin. The default value of 0 begins the tracing at the first DL/I call.

**STOP=**

Specifies a one- to eight-character hexadecimal value representing a count of DL/I calls. This value indicates the point at which DL/I image capture tracing



against a specific PSB should stop. The default value of X'7FFFFFFF' indicates that all of the DL/I calls for the batch program are traced.

**Defining DASD Logging Initialization Parameters**

In a DB/DC environment, use the OLDSDEF and WADSDEF control statements to specify the log data sets and parameters to be established during DLOG initialization. These control statements are mandatory and reside within the DFSVSMxx member of IMS PROCLIB. Use the ARCHDEF control statement to modify the GENJCL.ARCHIVE command generated by IMS when it switches OLDSs.

**Related Reading:** For additional information on archiving OLDSs and the GENJCL.ARCHIVE command, see “ARC=” under “Parameter Descriptions for IMS Supplied Procedures” on page 255. When you specify ARCHDEF in the DFSVSMxx PROCLIB member, it applies only to the active system. ARCHDEF specified in the DFSRSRxx PROCLIB member is meaningful only to a remote system.

**Control Statements:**

*ARCHDEF Statement:*

▶▶—ARCHDEF *aaaa*—▶▶

When IMS automatically archives OLDSs in response to the ARC= parameter of the IMS procedure, it does so by internally issuing a GENJCL.ARCHIVE command to DBRC. The default command is

GENJCL.ARCHIVE ALL SSID (xxxx)

Use the ARCHDEF control statement to change this command to:

GENJCL.ARCHIVE *aaaa* SSID (xxxx)

In this statement, *aaaa* represents any valid GENJCL parameter up to 63 characters in length.

*OLDSDEF Statement:*

▶▶—OLDSDEF—OLDS=(*id*), BUFNO=*nnn* MODE=*SINGLE*  
 DEGRADE=*YES*  
 NO

The OLDSDEF statement defines the OLDS data sets to be established during initialization. It consists of one or more 80-byte records placed within the DFSVSMxx member of IMS PROCLIB. If multiple OLDSDEF statements exist, all statements are processed. The OLDSDEF syntax is free-form, and continuation statements are indicated by a character other than a blank (X'40') in column 72. Keywords are delimited from associated parameters by an equal sign (=), and keyword/parameter pairs are delimited by a comma (,).

**OLDSDEF**

Identifies this as an OLDSDEF control statement. This keyword must appear first.

**OLDS=**

Identifies a list of OLDS to be allocated at DLOG initialization, where *id* indicates a list of two-digit OLDS IDs between 00 and 99. You can specify between 3 and 100 OLDS IDs. A minimum of three OLDS IDs is required.

**BUFNO=**

Identifies the number of log buffers allocated for OLDS read and write operations, where *nnn* is a one- to three-digit numeric integer between 002 and 999. The default is 005.

**MODE=**

Identifies whether the logger is to operate in SINGLE or DUAL logging mode. The default is SINGLE.

**DEGRADE=**

Identifies whether to degrade to single-logging mode (DEGRADE=YES), or to terminate IMS (DEGRADE=NO) after an OLDS write error. The default is DEGRADE=YES.

You should provide dynamic allocation members (via the DFSMDA macro) for all OLDS data sets. All (primary and secondary) OLDS must be preallocated. You must specify the OLDS blocksize when the data set is allocated.

*WADSDEF Statement:*



The WADSDEF statement defines the WADS data sets to be established during initialization. It consists of one or more 80-byte records placed within the DFSVSMxx member of IMS PROCLIB. If multiple WADSDEF statements exist, only the first encountered statement is processed. The WADSDEF syntax is free-form, and continuation statements are indicated by a character other than a blank (X'40') in column 72. Keywords are delimited from associated parameters by an equal sign (=), and keyword/parameter pairs are delimited by a comma (,).

**WADSDEF=**

Identifies this as a WADSDEF control statement. This keyword must appear first.

**WADS=**

Identifies a list of WADS to be allocated at DLOG initialization, where *n* indicates a list of one-digit WADS IDs between 0 and 9. You can specify between one and ten WADS IDs. A minimum of one WADS ID is required.

You should provide dynamic allocation members (using the DFSMDA macro) for all WADS data sets. All (primary and secondary) WADS must be preallocated. The WADS block size is extracted from the DSCB. The specification of WADS duplexing is provided via the WADS= EXEC parameter specified in the control region JCL (WADS=S for single WADS, WADS=D for dual WADS).

## Defining IMS Batch without Dynamic Allocation

This section describes control statements in the DFSVSMxx member that you use to disable batch dynamic allocation.

### **Control statements:**

►►—NODYNALLOC—◄◄

### **NODYNALLOC**

Disables batch dynamic allocation. Message DFS2480 appears at initialization time to indicate that batch dynamic allocation is not to occur. This statement disables the default setting, which is batch dynamic allocation enabled.

## Disabling Parallel Processing During Database Open

IMS uses multiple task control blocks (TCBs) to open and close full-function databases in parallel. To disable this default behavior and serialize database open processing under a single TCB, specify the NOPDBO option.

### **Control statements:**

►►—NOPDBO—◄◄

### **NOPDBO**

Disables parallel processing of databases during the OPEN process. The keyword NOPDBO must start in column 1. If this option is not specified, parallel processing is enabled by default.

## Defining Coupling Facility Structure Names for Sysplex Data Sharing

If you use sysplex data sharing, three coupling facility structure names must be passed to IMS:

- IRLM
- OSAM
- VSAM

These names are passed to IMS in the CFNAMES control statement in the DFSVSMxx member. The purpose of this statement is to let you select which of the structures, previously defined in the z/OS policy, is to be used at run time.

You can specify one or more CFNAMES control statements. Each statement can contain one or more keyword parameters. Each keyword parameter must be specified in its entirety on a single CFNAMES statement. A specific keyword parameter can not be continued from one CFNAMES statement to another. If keyword parameters are duplicated, the first parameter encountered is used and the others are registered with an error message.

The structure names that you specify in the CFNAMES control statement can be 1 to 16 characters long. All keywords in the statement must be coded, but null values for the CFOSAM and CFVSAM keywords are allowed.

**IRLM:** If you do not specify structure names in this PROCLIB member or specify only the IRLM structure name, the default is for the environment defaults to use two-way data sharing using the notify protocol.

If the IRLM and OSAM or VSAM structure names do not match the structure names known to the coupling facility at the time of the IDENTIFY, the IDENTIFY is rejected and IMS initialization fails. The first IMS in a data sharing group that identifies itself to the IRLM, sets the data sharing environment for all other IMS subsystems connecting to the same IRLM structure. The first IMS, in other words, sets the structure names for the coupling facility. If, for example, the first IMS doesn't specify structure names in this PROCLIB member, no other IMS subsystems identifying to this IRLM can specify structure names (with the exception of an IRLM structure name matching the one set by the IRLM); otherwise, the IDENTIFY is rejected and IMS initialization fails. The same thing happens if the first IMS specifies all three structure names and, later, another IMS subsystem tries to identify using one or more different structure names.

**OSAM:** If you are using OSAM sequential buffering, the OSAM structure registers SB buffers in the coupling facility's cross-system buffer invalidation process.

**Control Statements:** CFNAMES must appear in column 1. No blanks can appear between the keywords.

**CFNAMES=**

Specifies the coupling facility structure names that must be passed to IMS.

**CFIRLM=aaaaaaaaaaaaaaaa**

Specifies the coupling facility lock table structure name for the IRLM.

**CFVSAM=bbbbbbbbbbbbbbbb**

Specifies the coupling facility XI structure name for VSAM.

**CFOSAM=**

You can specify either the coupling facility name only, or the coupling facility name and directory-to-element ratios.

**cccccccccccccccc**

Specifies the coupling facility structure name for OSAM.

**(cccccccccccccccc,DIRRATIO,ELEMRATIO)**

Specifies the coupling facility structure name for OSAM and the directory-to-element ratio that the CF uses to configure the structure for data caching.

The DIRRATIO and ELEMRATIO sub-parameters are optional. The installation uses the CFRM policy to specify the size of a cached structure. When the structure is allocated, its storage is subdivided to reserve space for directory entries and data elements. The directory-to-element ratio determines the proportion of the cache used for directories and data elements. The ratio is expressed as a pair of whole numbers and is used when a connection is made to the structure. The directory-to-element ratio should reflect the average number of data elements per cache entry. The size of each data element is 2 KB. OSAM data is contained in the coupling facility as multiples of 2 KB data elements. If the ratio is incorrect for an installation's use of the structure, frequent rejections of cache requests can occur because either the cache or the cache structure is full.

The result of dividing the ELEMRATIO value by the DIRRATIO value must not exceed 16, the maximum number of elements supported by an OSAM structure. If the result is greater than 16, the specified ratio is rejected and IMS supplies a ratio.

The DIRRATIO value represents the directory part of the ratio and ELEMRATIO represents the element part of the ratio. Both sub-parameters

are coded as one to three digits. If you choose to specify one of the sub-parameters, you must specify both.

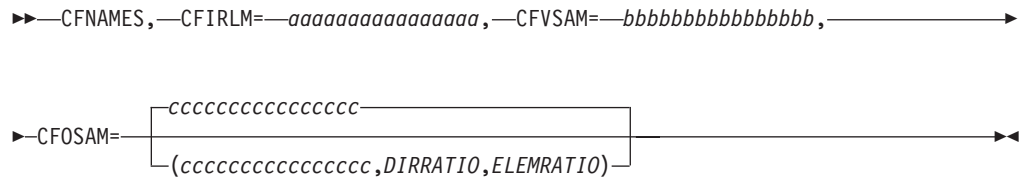
If you omit the DIRRATIO or specify a value of zero, IMS provides a default DIRRATIO of 1:0. If DIRRATIO specifies a non-zero value and ELEMRTATIO specifies zero, IMS supplies a ratio of 1:0. This ratio causes the coupling facility to configure the OSAM structure to preclude data caching. If data caching is precluded, the OSAM structure only supports buffer invalidation processing and does not support cache data.

If you omit the DIRRATIO and ELEMRTATIO subparameters, or if they fail validation of the maximum elements number, IMS provides a default ratio.

IMS supplies a default ratio of 999:1. This ratio causes the coupling facility to configure the data caching OSAM structure to optimize buffer invalidation and not data caching. While data caching might be possible, the structure provides limited space for data elements.

**Related Reading:** For additional information on sysplex data sharing, see *IMS Version 9: Administration Guide: System*.

Specifies the coupling facility XI structure name for OSAM.



### Using the Coupling Facility for OSAM Data Caching

The OSAM database coupling facility caching function allows you to optionally specify the caching of OSAM database buffers. For more information, see the description of the caching option (co) parameter under “Defining OSAM Subpools” on page 392.

When you specify the “cache only changed data” option and an application program modifies data in a subpool, changed data is written first to DASD then to the coupling facility. If you select the “cache all data” option and an application program requests data that is not already in a subpool or in the coupling facility, that data is either read from DASD or copied from the SB buffer into the subpool and then written to the coupling facility.

Performance varies, depending upon:

- The cache option you choose.
- The number of databases utilized.
- The database block sizes.
- The number of sharing IMS/z/OS images.

The “cache only changed data” option probably provides greater performance improvement than the “cache all data” option. If you specify the “cache only changed data” option, each data block from the selected database, once read from DASD, must be written to system memory. This action requires additional path length and cycles. The specific use of the database by your application program dictates whether this option is beneficial.

Data caching initialization processing is dependent upon the proper environment and the specification of the caching option in the subpool definition statement. The following actions are performed:

- Establish an element size of 2K for the coupling facility.
- Allocate a buffer prefix extension for caching subpools. The extension resides as a non-contiguous area and is anchored from the prefix. It consists of two parts:
  - A fixed length section for asynchronous processing.
  - A variable length section for data transfer buffer lists. The number of buffer lists is dependent upon the subpool buffer size.

**Example:** A subpool buffer size of 6 KB will require three buffer lists and a subpool size of .5 KB, or 1 KB will require one buffer list. A single buffer list will accommodate 2 KB of subpool buffer.

- Establish miscellaneous values for coupling facility data transfer parameters including:
  - Element and directory ratios
  - Buffer increment number (number of 256 byte segments)
  - Number of buffers in the buffer list.

The write data option is supported at the subpool level and permits the definition of each subpool with a write cache option.

**Restriction:** When using sequential buffering and the coupling facility for OSAM data caching, the OSAM database block size must be defined in multiples of 256 bytes (decimal). Failure to define the block size accordingly can result in ABENDS0DB PIC15 from the coupling facility. This condition exists even if the IMS system using sequential buffering is accessing the database in read-only mode.

### Enabling the Long Busy Handling Function

RAMAC disk arrays sometimes cause a problem to online programs when the DASD subsystem takes a very long time to do error recovery. This condition (known as *long busy*) might range in time from a couple of seconds to many minutes. You can avoid these excessive long wait times by defining the Multiple Area Data Sets I/O Timing (MADSIOT) keyword on the DFSVMxx PROCLIB member, thus enabling IMS's long busy handling function.

For more information on the long busy function, see "Managing I/O Errors and Long Wait Times" in *IMS Version 9: Administration Guide: Database Manager*.

#### Control Statement:

►►—MADSIOT=(*structurename*,*iotime*)—◄◄

<i>structurename</i>	Specifies the name of the coupling facility list structure that IMS will use for the long busy handling function.
<i>iotime</i>	Specifies the maximum time (in seconds) before the long busy handling function is activated. The allowable values are 0 to 255. This value will be passed to the I/O subsystem for all I/O requests to MADS. If the time interval specified expires before the request completes, the request is terminated.

This explicit request timing value takes precedence over an I/O timing value specified at the volume level.

### Enabling the IRLM Lock Timeout function

The IRLM Lock Timeout function allows you to interrupt processes that are waiting for locks for longer than a specified number of seconds. To use this function, specify a decimal integer value for the LOCKTIME parameter in the DFSVSMxx IMS.PROCLIB member (or DFSVSAMP DD statement for IMS batch procedures). The LOCKTIME parameter specifies the number of seconds that IRLM waits before rejecting lock requests that have not been granted. For example, when LOCKTIME=10 is specified, IRLM waits ten seconds before rejecting lock request that have not been granted. The LOCKTIME parameter is only supported when using IRLM 2.1. or higher.

When no value is specified for the LOCKTIME parameter, IRLM issues a message DXR162I to indicate that a task has held (or has been waiting for) a lock for longer than five minutes. RMF records (type 79.15) can be formatted for more information about the task that is holding (or waiting for) the lock.

When a value is specified for the LOCKTIME parameter, IRLM interrupts all of the dependent regions that have been waiting for a lock for longer than the number of seconds specified by the LOCKTIME parameter. An ABENDU3310 is issued for every dependent region that waits for longer than the number of seconds specified by the LOCKTIME parameter.

#### Control Statement:

```
►►—LOCKTIME=value—————►►
```

*value*

Specifies the number of seconds that IRLM is to wait before rejecting lock requests that have not been granted. Locks that are already granted cannot be rejected. Valid values for this parameter are from 1 to 3275.

## Member DFSYDTx

The DFSYDTx member of IMS.PROCLIB specifies the OTMA descriptors that are built during IMS initialization. OTMA descriptors are optional. All parameters are delimited with a blank. The following is the format of a descriptor:

Column	Contents
1	Descriptor type. M for an OTMA descriptor.
2	Blank.
3-18	One- to sixteen-character client name, left-justified and padded with blanks, if necessary. This is a required and positional parameter.  Duplicate names are not allowed.
19	Blank.
20-31	Keyword: <b>DRU=</b> OTMA Destination Resolution exit routine name. This is a required parameter.

Duplicate exit routine names are allowed.

**73-80** Sequence numbers. These columns are ignored by IMS.

Up to 50 lines can be used in the specification of a descriptor. Columns 1 through 18 must be the same for each line.

## Member DFS62DTx

LU 6.2 device descriptors are built during IMS initialization. They reside in member DFS62DTx in IMS.PROCLIB, where *x* is the IMS nucleus suffix. LU 6.2 device descriptors specify an LTERM that associates an output destination with an LU 6.2 device. The system administrator can additionally change applications that use alternate PCBs into applications that use LU 6.2 devices, without application coding changes.

No IMS definition or system definition changes are required to use an LU 6.2 device.

The record format is:

Position	Contents
1	Descriptor type ("U" for user descriptor)  Although this column is not required for an LU 6.2 descriptor, it is included for consistency with ETO descriptors. It is not checked, and no error message is issued if it is omitted.
2	Blank
3-10	Contains the eight-character LTERM name, left-justified and padded with blanks, if necessary. This positional parameter is required, and is used as the descriptor name. A descriptor definition can be continued by repeating the descriptor name in subsequent records. Each individual descriptor can contain up to 50 records (excluding comment records).
11	Blank
12-72	Contains the parameter sets for the descriptor.
73-80	Ignored

### DFS62DTx Descriptor Format

This is the format for the DFS62DTx descriptor. Do not code a parameter in the descriptor and then leave it blank such as (SIDE=b). If you do this, an error message is issued. Instead, omit it completely if you don't need to specify a DFS62DTx descriptor. A descriptor definition can be continued by repeating the descriptor name in subsequent records.

```
A ltermname parm1 parm2 parmn
```

where

U

is for user descriptor

ltermname

is the LTERM name

parm (1...n)



can be any of the following parameters: SIDE= MODE= TPNAME= SYNCLEVEL=  
CONVTYPE=

Column	Column	Column
1	12	72
U DESC1	LUNAME=SOMELU	SIDE=SOMESIDE MODE=SOMEMODE CONVTYPE=SOMECONV
U DESC1	TPNAME=SOMETP	

Separate the parameters with a blank.

## Keyword Descriptions

### SIDE=

Is a one- to eight-character alphanumeric name identifying side information entry. (Side information refers to the system-defined values for CPI communications initialization.) If you specify an entry name on this keyword, the LUNAME, TPNAME, and MODE keywords are blank unless you also specify an overriding LUNAME, TPNAME, or MODE. If you do not specify an entry name, the default values apply to the LUNAME, TPNAME, and MODE keywords.

### LUNAME=

Is a 1- to 17-alphanumeric character (uppercase) name of the destination LU 6.2 application program, which can be a 1- to 17-byte network-qualified LU name. An example of a network-qualified LU name is *netid1.luname1*. If SIDE= is specified, LUNAME overrides the LUNAME in the side information entry. The default is DFSLU.

Do not specify a network name for a dependent LU.

### TPNAME=

Is a 1- to 64-character name of the transaction program to be scheduled. The default is DFSASYNC. If SIDE= is specified, TPNAME overrides the TPNAME in the side information entry.

**Recommendation:** Use upper-case characters for TPNAMEs so that the /ALLOCATE and /CHANGE commands can operate on the resulting messages and descriptors.

### MODE=

Is a one- to eight-character name of the VTAM mode table entry to be used. If SIDE is specified, MODE overrides the mode name in the side information entry. The default is DFSMODE.

### SYNCLEVEL=

Specifies whether the APPC/IMS sync level is confirmed (C) or not (N). The default value is C.

### CONVTYPE=

Specifies whether the conversation type is basic (B) or mapped (M). The default value is M.

---

## Specifying IMS Execution Parameters

You can specify control region execution parameters in the appropriate control region JCL, or in the IMS, DBC, and DCC PROCLIB members. Parameters specified in these members override any system definition. The parameters are not positional.

## Database and PSB Exec Parameters for the Control Region

Three kinds of parameters in the IMS procedure that are related to database and PSB processing are:

- Pointers to database buffer requirements
- Overrides of predefined buffer sizes
- Performance options

### Database Buffer Requirements

Use the VSPEC parameter to specify a two-character suffix for member DFSVSMxx in IMS.PROCLIB. This member predefines the buffer pool requirements for databases that use OSAM or VSAM as the access method. The IMS procedure generates a default value of 00 that points to a general-purpose buffer definition in member, but you can change this suffix.

See “Specifying IMS Buffer Pools” on page 430 to specify subpool sizes.

### Buffer Sizes

Several parameters enable you to redefine the size of buffer pools that hold DMBs and PSBs. Specify a value that represents the size as a number of 1024-byte blocks. For example, a value of 18 represents the size as 18432 bytes (18 x 1024 = 18432). IMS then rounds up this number based on the nearest whole number of pages on which these bytes fit. For example, if one page in z/OS equals 4096 bytes, the value of 18 blocks would result in the allocation of 20480 bytes. This value overrides the value defined in system definition. The calculation is as follows:

#### Formula:

$$18 \times 1024 = 18432 \text{ bytes} \\ 18432 / 4096 = 4.5 \text{ pages (round up to 5)} \\ 5 \times 4096 = 20480 \text{ bytes}$$

Performance studies often indicate a need to increase DMB or PSB buffer space. You can override the system definition values to implement this change.

If you use the DL/I address space option, two PSB pools exist: one in the z/OS common area and one in the DL/I address space. Use the SASPSB parameter on the BUFPOOLS macro to specify sizes of the PSB pools during IMS system definition; use the CSAPSB and DLIPSB parameters on the IMS procedure to override the size of the PSSB pools. If the DL/I address space is not used, specify the size of the single PSB pool using the PSB= parameter on the BUFPOOLS macro; override this value with the PSB= parameter on the IMS procedure.

If Fast Path resources are used, an EPCB pool must contain extensions to Fast Path PCBs. For each PCB referencing a DEDB or MSDB database, an EPCB is required. This pool is used only by MPP regions, not by IFP or BMP regions.

Use the parameters of the IMS procedure in Table 46 to override the buffer pool sizes that were specified at system definition time with the BUFPOOLS macro.

Table 46. Overriding Buffer Pool Sizes with Database and PSB Parameters

IMS Procedure Parameter	System Definition Specification	
	Macro	Parameter
DMB	BUFPOOLS	DMB
DBWP	-	-
PSB	BUFPOOLS	PSB
PSBW	BUFPOOLS	PSBW
CSAPSB	BUFPOOLS	SASPSB(size1)
DLIPSB	BUFPOOLS	SASPSB(size2)

Table 46. Overriding Buffer Pool Sizes with Database and PSB Parameters (continued)

IMS Procedure Parameter	System Definition Specification	
	Macro	Parameter
EPCB	BUFPOOLS	EPCB

The IMS procedure is generated with null value parameters for these storage areas. The sizes default to their system definition values.

### Performance Options

System definition lets you specify individual database DMBs and PSBs to be resident with the DATABASE and APPLCTN macros, respectively. You can override this specification with the RES parameter. The IMS procedure generates this as a null parameter; the default is RES=Y.

Request that all the control blocks are not to be made resident by specifying RES=N. You might do this when executing a test system or if storage is temporarily constrained.

## Data Communication EXEC Parameters for the Control Region

Three kinds of parameters in the IMS procedure are related to data communication:

- Overrides of predefined buffer sizes
- Overrides and options particular to Message Format Service (MFS)
- Performance options

### Buffer Sizes

By monitoring communication traffic you might decide to increase buffer space. Use the parameters of the IMS procedure in Table 47 to allocate virtual storage and override the buffer pool sizes predefined in system definition.

Table 47. Overriding Buffer Pool Sizes with Data Communication Parameters

IMS Procedure Parameter	System Definition Specification	
	Macro	Parameter
FBP <sup>1</sup>	BUFPOOLS	FORMAT(size1)
QBUF <sup>2</sup>	MSGQUEUE	BUFFERS
RECA	COMM	RECANY

#### Note:

<sup>1</sup> Specify a value representing the number of 1024-byte blocks.

<sup>2</sup> By increasing the number of buffers (used for the message queue data sets), you can reduce the frequency of I/Os.

### MFS Options

When you use MFS, you can adjust the capacity of the system by adjusting the availability of format blocks. Use the FRE parameter to specify the maximum number of active blocks in the message format buffer pool. You need to coordinate this number with the value specified for the FBP pool size parameter.

### Performance Options

The VAUT parameter offers a performance option to systems using VTAM. You can override the null value generated in the IMS procedure and specify the use of VTAM Authorized Path.

The NLXB parameter allows you to add parallel sessions during system startup. The number you specify is added to the number defined in the SESSION keyword of the MSPLINK macro.

The FESTIM parameter overrides the timeout value for front-end switching that is predefined in system definition. You specify the parameter on the COMM macro with a value from 1 to 300 seconds.

## System Control and Performance EXEC Parameters for the Control Region

The positional parameters that affect control of the system resources follow.

### Identification of Nucleus

The SUF parameter is generated as a null value for the IMS procedure, indicating that the default nucleus name has a one-character suffix of 0. If you are using an alternative control program, you must specify this parameter.

### Active Regions

Use the PST parameter to override the expected number of regions that are to be in operation during the online execution. Additional regions can be dynamically allocated, up to the maximum allowable number permitted by your operating system. If you do not specify the PST parameter, the default number of regions is the number specified for the MAXREGN keyword in the IMSCTRL macro.

### Performance Options

Three parameters contribute to general performance strategy. The FIX parameter specifies a two-character suffix for the member DFSFIXxx in IMS.PROCLIB. This suffix indicates which member should describe all modules and control blocks that are to be page fixed. The EXVR parameter allows you to page-fix buffers used for the management of message queues. The parameter is generated with a null value; you must specify EXVR=1 to indicate the page-fixing action. Similarly, the PRLD parameter specifies a two-character suffix for the member DFSMPLxx, where all preloaded modules are listed.

### z/OS Options

Several parameters apply only to the z/OS operating system. The SRCH parameter allows you to take advantage of any special library structure to optimize the search for loaded modules. You can override the default value of 0 with a value of 1 if you want the JPA and LPA to be searched before IMS program libraries.

To reduce the amount of z/OS CSA (common storage area) that IMS uses you can use one of the following techniques:

One technique is to specify LSO=Y on the IMS procedure to cause some control blocks and some IMS modules that are used for DL/I processing to be loaded into the control program's private storage. To allow for this transfer, use the following sum to increase the size of the control region:

#### Formula:

220K + OSAM buffer pool + VSAM buffers  
+ enqueue/dequeue tables + system log buffers

Another way to reduce IMS use of CSA is to take advantage of log buffer CSA VSCR (virtual storage constraint relief). Using CSA VSCR, several buffers can be moved out of CSA and placed above the 16-megabyte line into ECSA (extended CSA). The buffers that are relocated are the online log buffers, the batch log buffers, and the IMS Monitor buffers. This method not only frees CSA space, but it also allows these buffers more space in ECSA. To use this buffer relocation technique, specify LSO=S on the IMS procedure.

For details about specifying values on the LSO= parameter, see the description of the LSO= in Chapter 4, "Procedures," on page 191.

For operations, you can specify the output class of a spinoff dump using the SOD parameter. It is strongly recommended that you override the null value generated for this parameter for all z/OS installations; it provides immediate analysis of the status of real storage after abnormal termination of the control region or of each dependent region.

### DL/I Address Space

An additional variation of the local storage option is to use the DL/I separate address space. You do this by specifying LSO=S. This address space contains most of the DL/I code, control blocks, and database buffers for full-function databases. Again, z/OS cross memory services are used.

### Dump Formatting Options

In a DC environment, you can request the following types of dump outputs for errors that terminate IMS: SDUMP, SYSMDUMP, SYSABEND, or SYSUDUMP. To do this, specify the FMTO startup parameter in combination with z/OS dump DD statements.

For SYSMDUMP, you should provide operational procedures for saving and formatting dumps; otherwise, you can overlay a SYSMDUMP if you must restart IMS before the previous SYSMDUMP is transferred.

You can also request dump outputs for some errors that do not terminate IMS. Your choice of dump depends on several factors: the type of failure, the FMTO parameter option, and the IMS spinoff and z/OS dump DD statements that have been selected. For more information about the FMTO parameter and about using dumps in these situations, see Chapter 6, "Installing and Tailoring the Interactive Dump Formatter," on page 315.

### IRLM Options

If an execution of the IMS online system is to use IRLM as lock manager or participate in block-level sharing, specify IRLM=YES. Specify the z/OS subsystem name for the IRLM associated with this control region on the IRLMNM parameter.

**Related Reading:** For details of IRLM requirements, see *IMS Version 9: Administration Guide: System*.

If your online system includes Fast Path, use the UHASH parameter with the IRLM to specify the name of an alternative hashing module. You must specify the appropriate name if you did not use the default name of DBFLHSH0 in system

definition. You cannot change this name across a restart. If multiple IMS subsystems are participating in data sharing and are using this option, the module must be identical in all systems.

### Subsystem Identification Parameters

Use the SSM parameter to reference a member in IMS.PROCLIB which identifies the DB2 subsystems that can be accessed from application programs executing in dependent regions. The one- to four-character suffix that you specify, together with the currently assigned name for IMSID, forms the member name. The member contains entries, each identifying a DB2 subsystem (by its z/OS subsystem name). All subsystems to be accessed from programs executing in dependent regions must be identified by an entry.

Depending on its entry, a region can access all, some, or none of these subsystems. To allow a dependent region to access all subsystems identified to the IMS control region, do not specify the SSM parameter for the dependent region or have the SSM entry specify the same member as the IMS procedure. To allow a dependent region to access only selected DB2 subsystems, specify an SSM parameter on the dependent region procedure that points to a member containing only those specific subsystems. To prevent a dependent region from having access to any subsystem identified to the IMS control region, specify a member containing no entries.

See Chapter 11, "Accessing External Subsystems with IMS," on page 475 for details about how to specify the individual entries and construct the IMS.PROCLIB member.

## Recovery-Related EXEC Parameters for the Control Region

### DBRC

Batch and utility regions use the DFSIDEF0 module during initialization. The DFSIDEF0 module that is shipped in the ADFSSMPL library contains a batch and utility region default of DBRC=YES. This value is coded on the DFSIDEF macro.

You are not required to use the DFSIDEF0 module. If you do not use the module or if the module cannot be loaded at initialization time, IMS defaults to DBRC=YES. Therefore, if DBRC is to be used for your batch and utility regions, then creating the DFSIDEF0 module is not required.

However, you may want to change the current value for DBRC. You can do this in one of three ways:

- Assemble and bind DFSIDEF0 into the IMS execution library
- Override the value in a DFSPBxxx proclib member.
- Override the value in JCL.

Figure 33 on page 421 is sample JCL to do the assembly and bind of DFSIDEF0:

```

//ASSEMBLE EXEC PGM=ASMA90,PARM='NOOBJ,DECK'
//SYSLIB DD DSN=IMS.SDFSMA,DISP=SHR
//SYSPUNCH DD DISP=OLD,DSN=IMS.OBJDSET(DFSIDEF0)
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,DISP=(,DELETE),SPACE=(CYL,(15,15))
//SYSIN DD *
IDEF0 TITLE 'DFSIDEF0 - IMS INSTALLATION DEFAULTS BLOCK'
DFSIDEF0 CSECT
        SPACE 1
        DFSIDEF TYPE=BEGIN
        DFSIDEF TYPE=PARAM,DBRC=YES
***** DFSIDEF TYPE=PARAM,DBRC=NO
***** DFSIDEF TYPE=PARAM,DBRC=FORCE
        DFSIDEF TYPE=END
        END DFSIDEF0
//STEP1 EXEC PGM=IEWL,
//          PARM='SIZE=(880K,64K),NCAL,LET,REUS,XREF,LIST'
//SYSPRINT DD SYSOUT=*
//SYSPUNCH DD DSN=IMS.OBJDSET,DISP=SHR
//SYSMOD DD DSN=IMS.SDFSRESL,DISP=SHR
//SYSUT1 DD UNIT=(SYSDA,SEP=(SYSMOD,SYSPUNCH)),SPACE=(CYL,(10,1))
//SYSLIN DD *
INCLUDE SYSPUNCH(DFSIDEF0)
NAME DFSIDEF0(R)

```

Figure 33. Assembly and bind of DFSIDEF0 module

### WADS=

The WADS parameter indicates whether the IMS online system is to use a single write-ahead data set (WADS) or dual data sets. The WADS helps protect the integrity of the online logging. In case of write errors, additional data sets can be defined up to a total of ten. Specify WADS=S for single recording or WADS=D for dual recording.

### ARC=

Use the ARC parameter to specify whether automatic archiving of the online log data sets (OLDS) is to be performed. Automatic archiving is recommended; however, you can arrange for the MTO to monitor the availability of the OLDS and perform archiving when necessary. Specify ARC=0 for no automatic archiving; otherwise, specify a one- or two-digit number representing the number of full (closed) OLDSs that invoke automatic archiving for those data sets.

### DBRCNM=

Specify a name (do not accept the default) for the start procedure for the DBRC region on the DBRCNM parameter. The default is a null parameter indicating a procedure name of DBRC. Also as part of your installation's recovery strategy, the AUTO parameter specifies whether to use automatic restart of the IMS online system. You need to override the generated default value (N) by specifying AUTO=Y if you plan to use automatic restart.

### QTU= and QTL=

For a non-shared queues environment, the QTU and QTL parameters are associated with protection of the message queues. Use them to adjust the values in the message queue space notification exit routine (DFSQSPC0). Given the number of records reserved on each message queue data set for automatic shutdown (specified in system definition with the SHUTDOWN parameter on the MSGQUEUE

macro), a finite number of residual records are available for message use. The notification exit routine keeps track of the current level of utilization. If the usage exceeds the upper limit (expressed as a percentage of records used), the MTO is notified so that steps can be taken to reduce the backlog. The MTO is also notified if the usage falls below the lower limit.

For a shared queues environment, the QTU and QTL parameters are used to monitor the DRRN in-use count. If the high threshold value is reached, a DFS2281I message is issued, and all messages retrieved from shared queues are put in an IWAITstate. When the DRRN in-use count becomes less than low threshold value, a DFS2282I message is issued, and the messages in an IWAITstate are then posted.

The QTU and QTL values are one or two-digit percentages. QTU overrides the upper threshold value and QTL overrides the lower one (100 is allowed for QTU). Both upper and lower threshold values are generated with null values; the defaults are those of the IBM-supplied exit routine—75% and 60%, respectively. Unless your installation establishes its own exit routine to control the MTO notification, you should probably use these defaults. In the long term, you can tune them to fixed values based on the feedback from the statistics given in the /DISPLAY POOL command; enter this command during peak periods of activity.

**Related Reading:** Details of the IBM-supplied notification exit routines are in *IMS Version 9: Customization Guide*.

## Security-Related EXEC Parameters for the Control Region

Several parameters control the kind of security checking that is done during the current execution by acting as switches for different types of security; moreover, they determine how much flexibility the MTO has to override the choice of security checking. You must coordinate the setting of these switches with both overall security design and operational procedures. The parameters are TRN, SGN, RCF, and ISIS.

All the values generated for the IMS procedure parameters specify **no** security. You need to reset them to match the security design. You must also coordinate the level of the security tables with the suffix identifier for the nucleus.

**Related Reading:** Operational restrictions for the MTO, and the parameter values and their meaning are discussed in *IMS Version 9: Administration Guide: System*.

The IMSID parameter is partially related to security and to operations. The one- to four-character value that you specify uniquely identifies the control region. Dependent regions executing under control of this nucleus must specify the same identifier. The definition default is IMSA.

Any message sent to the console from subsystem execution is identified by this name; therefore, the name chosen should be unique to other subsystems executing in the z/OS system, including any batch executions. The IMSID name should not be the same as the procedure name in order to avoid confusion as to the source of the message.

## IMS Message Processing Region Parameters

This section does not apply to DBCTL.

The PARM= positional parameters for the message processing region's EXEC statement are shown in Table 48. See the DFSMPR procedure for more information.



Table 48. Categories and Purpose of Message Processing Region Parameters

Category	Parameter	Purpose
Database and PSB	PCB	Specify size of pool for PSB copy
	VALCK	Check validity of addresses in DL/I calls
	NBA	Specify Fast Path database buffers
	OBA	Specify Fast Path overflow buffers
Data Communication	CL1	Specify priority of message class
	CL2-4	Specify up to three lower-priority message classes
Region Control and Performance	LOCKMAX	Turn off locking limitations
	MSG	Specify MSG for message region
	OPT	Coordinate region start with control region
	OVLA	Retain overlay supervisor
	PRLD	Specify suffix of DFSMPLxx for program preload
	PREINIT	Specify suffix of DFSINTxx for modules to be preinitialized in readiness for takeover
	STIMER	Specify timer options
	DBLDL	Specify maximum number of IMS.PGMLIB directory entries to be saved
	SOD	Specify output class for spinoff dump
	SSM	Specify access to DB2 subsystems
	ALTID	Specify alternate IMS system
	PWFI	Activate pseudo wait-for-input
	APARM	Specify application parameter
Recovery and restart	SPIE	Allow user program SPIE during DL/I call
	TLIM	Specify number of abnormal terminations allowed
Security options	AGN	Specify application group name
	IMSID	Override IMS subsystem identifier

### PSB-Related EXEC Parameters for a Message Processing Region

The PCB parameter is an optional parameter that specifies the size of the interregion communication area. This area holds the PCBs included in the PSBs that are used by programs executing in this region. If you do not specify this parameter, the default value is 000. This allocates an area large enough to hold all the PCBs for the largest PSB defined to this IMS system and present in the active IMS.ACBLIBA/B. To specify a value, enter the number of 1024-byte blocks as a three-digit number (for example, 001).

The default value for the VALCK parameter signifies that address validity checking is not to be performed for DL/I calls issued by the application programs in this region. (An address is invalid if it is either lower than the lowest address not in the

z/OS nucleus or higher than the highest address in virtual storage.) With adequate testing of and controls over the DL/I call parameter coding, validity checking should not be necessary.

### **Data Communication EXEC Parameters for a Message Region**

A message region requires the set of four parameters, CL1, CL2, CL3, CL4, to specify transaction classes. Express each class as a three-digit number. The first message class for the region causes all messages assigned to that class to be selected first as eligible for scheduling. Only when all possibility of scheduling a transaction in that class has been exhausted does scheduling begin for the second message class. Priorities determine the order that programs are to be selected for scheduling into the region. The message classes you specify need to be coordinated with your transaction scheduling rules and the numbers entered with the PRTY keyword on the TRANSACT macro.

### **Region Control EXEC Parameters for a Message Region**

MSG is the first positional parameter for a message processing region. When transactions are encountered on the queue, programs are automatically scheduled in these regions, if the message class priority is suitable.

The parameter OPT= helps you control the region startup. If, for some reason, the control region is not active or is terminating when the MPP region is invoked, you can have the MTO decide whether to start the MPP region again (the default), let it wait until the control region is ready, or cancel it.

Use STIMER= to invoke timer facilities. If you use the non-default LSO option, LSO=S, both application program and DL/I processing times are recorded. Most DL/I processing time is not included when you use the default, LSO=Y. A value of 1 causes a timer sequence to be issued once for each DL/I call that can adversely affect the performance of the region. This parameter is used in conjunction with the TIME= parameter on the //JOB statement. Because a timeout of a message region causes an abnormal termination of the message region, it is recommended to have processing limits coded within the application program.

Use the SSM= parameter, which is required when using DB2 group names to point to a member in IMS.PROCLIB that identifies the DB2 subsystems that can be accessed from this MPP region. To prevent any access to subsystems from an MPP scheduled into this region, use the name of a null member (no entries). The SSM= default, a null value, allows the MPP region to attach to any of the subsystems declared to the IMS control region. If necessary, coordinate this parameter with the corresponding SSM parameter in the IMS procedure. Refer to the description in "System Control and Performance EXEC Parameters for the Control Region" on page 418.

Use the APARM= parameter to specify execution time parameters unique to this dependent region. This parameter specifies a character string for the application program or Data Capture exit routine. The APARM= can be used to specify the frequency for checkpoint calls. The APARM= can also be used to pass parameters to the Data Capture exit routine to indicate whether the data should be captured.

The maximum length of the parameter is 32 characters; the parameter must be enclosed in single quotes (') if special characters are used; embedded commas are not allowed. The INQY call with the ENVIRON subfunction is used to receive the APARM character string.

**Related Reading:** For more information, see *IMS Version 9: Application Programming: Database Manager*.

Use the SOD parameter to specify the output class for a spinoff dump. In this way, you can obtain a printed copy of the storage immediately at abnormal termination of the message region immediately rather than being placed on general SYSOUT queues.

### Performance-Related EXEC Parameters for a Message Region

Four performance-related options are available. The parameter OVLA= enables you to retain a copy of the overlay supervisor in the message region. The default is to have the overlay control loaded each time you schedule an overlay application program in the region. If you have arranged the message priorities so that an overlay program is likely to be scheduled in the region, you should use this option.

Use the DBLDL= parameter to minimize the cost of program loading by increasing the number of program directory entries maintained in the message region. The effect is to reduce the I/O to the program library (and the search) to obtain the direct address of the program. The default is 20.

Use the PRLD= parameter to specify the two-character suffix of the member DFSMPLxx in IMS.PROCLIB. This member contains the list of programs that are to be preloaded. The effect is to reduce the repetition of I/O for program load. The region address space retains the modules.

PREINIT allows you to specify the two-character suffix of the member DFSINTxx in IMS.PROCLIB. This member contains the preinitialization modules to be received control.

The type of program load selected depends on the order of the following:

1. Preload list
2. Standard z/OS fetch

The VFREE= parameter instructs IMS to release or retain virtual storage at program termination. If you use the program infrequently, and you need dependent region virtual storage, it is recommended that you specify a value Y for the VFREE= parameter to free the storage. If you are optimizing for programs that are not used continuously by the message processing region, the combination of freeing virtual storage is appropriate.

The fourth performance-related parameter is the PWFI= (Pseudo WFI) parameter. Specifying PWFI=Y can potentially reduce CPU time by eliminating the termination and rescheduling of resources. On a message GU call if MODE=SNGL is specified on the TRANSACT macro and no more messages are available, IMS checks to see if other work needs to be done for the region. If no other work is available and the resources owned by the region are not needed by another region with work to do, the region becomes idle and waits until another message is queued (wait-for-input mode). It does not return a "QC" status code to the application program. If the new message is for the transaction that is scheduled in the region, IMS returns the new message to the application program without having to terminate and reschedule the resources.

If the next message that comes in is not for the transaction that is scheduled in the region, but it can be processed by the region, there is a delay before the new transaction can be processed. Because termination of the previous message is

delayed until after the new message comes in, the region must go through termination before it can schedule the new transaction.

### Recovery-Related EXEC Parameters for a Message Region

The TLIM parameter addresses a problem with an application program that causes an abnormal termination. Because the program can be scheduled many times into a region as a result of transactions in the queue, you need to be able to stop the operation of this region. The value for TLIM specifies the limiting number of abnormal terminations permitted. In the case where the application program has a SPIE in effect, the SPIE option allows it to be left on during the DL/I call or for it to be turned off during the DL/I call and reinstated when returning to the application program. For performance reasons, it is not recommended to turn it on and off for each DL/I call. With PL/I Release 5 or later, you can use the PL/I SPIE facility without having IMS reset the SPIE on each DL/I call.

### Security-Related EXEC Parameters for a Message Region

The AGN parameter enables you to specify the application group name (AGN), that is associated with a set of transactions, PSBs, or LTERMs that are authorized to be used by this region. Declare these resources and the AGN in the Security Maintenance utility input to authorize them to be used. Include the same name to invoke IMS application group name (AGN) security for this region. The DFSMPR procedure generates this as a null parameter. If other regions are being controlled through the use of AGNs, you risk permitting unauthorized access by programs executing in this region if you omit this parameter.

The parameter IMSID is related to security and to operations. You specify the 1- to four-character identifier for the name of the control region (the name given for the IMSID parameter). If the value does not match the current IMSID of any operating control region, the message region is not scheduled.

## IMS Batch Message Processing Region Parameters

The PARM= positional parameters for the BMP region's EXEC statement are shown in Table 49. See the IMSBATCH procedure for more information.

Table 49. Categories and Purpose of BMP Region Parameters

Category	Parameter	Purpose
Database and PSB	MBR	Specify name of batch message program
	PSB	Specify PSB name
	TEST	Check validity of addresses in DL/I calls
	NBA	Specify Fast Path database buffers
	OBA	Specify Fast Path overflow buffers
Data Communication	IN	Specify an input transaction queue
	OUT	Specify an output transaction queue
Region Control and Performance	BMP	Specify BMP for batch message region
	OPT	Coordinate region start with control region
	LOCKMAX	Specify locking limitations
	PRLD	Specify suffix of DFSMPLxx for program preload
	PREINIT	Specify suffix of DFSINTxx for modules to be preinitialized in readiness for takeover

Table 49. Categories and Purpose of BMP Region Parameters (continued)

Category	Parameter	Purpose
	STIMER	Specify timer options
	CPUTIME	Specify processing time limit
	PARDLI	Specify region for DL/I processing
	DIRCA	Specify interregion communication size
	SSM	Specify access to DB2 subsystems
	ALTID	Specify alternate IMS system
	APARM	Specify application parameter
Recovery and Restart	SPIE	Allow user program SPIE during DL/I call
	CKPTID	Specify checkpoint for program restart
Security options	AGN	Specify application group name
	IMSID	Override IMS subsystem identifier

### PSB-Related EXEC Parameters for a Batch Message Region

The MBR= parameter specifies the name of the program and is often the same as the PSB name. This parameter is required. The BMP has flexibility in using a program and PSB combination. This allows you to test modifications of the BMP using a temporary program name. You can also use a different PSB with the same program.

The PSB= parameter is optional if it matches the MBR name. If an APPLCTN macro has been included for the PSB specifying BATCH as the program type, the PSB parameter specifies one of several PSBs for a given BMP. You should identify application programs that have the ability to process several types of transactions. These programs often require larger amounts of virtual storage, and you can need to adjust the size of the region.

The TEST= parameter is required, but the IMSBATCH procedure generates a 0 to specify that validity checking of the addresses in a DL/I call is **not** performed. (An address is invalid if it is either lower than the lowest address not in the z/OS nucleus or higher than the highest address in virtual storage.) Adequate testing of the program and controls over the DL/I call parameter list coding should make the generated option (no validity checking) appropriate.

The NBA= parameter specifies the number of Fast Path database buffers to be made available. These buffers are taken from the global pool defined by the DBBF= parameter in the IMS control region procedure.

The OBA= parameter specifies the number of additional database buffers to be made available in the CSA/ECSA. Fast Path allocates only the largest OBA specification from all currently active regions.

### Data Communication EXEC Parameters for a Batch Message Region

For the batch message program, you have the flexibility of declaring that the input transaction queue is to be made available to the program at execution time. You do this by specifying one transaction code as the value for the IN parameter.

Some BMP programs do not access a message queue; however, they have a requirement to send output to a terminal or to generate transactions to be

processed by other application programs. Specify the LTERM name or transaction code (as appropriate) on the OUT parameter.

A null value for the IN parameter prevents the program from accessing message queues. When you specify the transaction code for the IN parameter, the program has no restrictions on generated transactions or output messages.

### Region Control EXEC Parameters for a Batch Message Region

BMP= is the first positional parameter for a batch message processing region. These regions are not scheduled automatically. They must be invoked by the operator.

The OPT= parameter helps you control the start of a batch message region. If the control region is not active when the BMP region is invoked, you must decide whether to wait for a control program, cancel the batch message region, or ask the operator to make the decision. This might occur when JCL in the z/OS job stream for the BMP region is scheduled before the control region has completed its initialization or when the control region is terminating. Specifying 'wait' is a risk because the z/OS resource is reserved until the control region resumes. If you are starting the region from the master terminal, the default generated for the IMSBATCH procedure is satisfactory.

The optional PRLD= parameter specifies the two-character suffix for the member DFSMPLxx in IMS.PROCLIB, which lists the preloaded modules. It is optional; it is generated as a null parameter for the IMSBATCH procedure. For BMP regions that do not have other programs scheduled in them, PRLD provides no performance gain.

The PREINIT= parameter specifies the two-character suffix of the member DFSINTxx in IMS.PROCLIB. This member contains the preinitialization modules that are to receive control.

The optional STIMER= and CPUTIME= parameters are generated as null parameters, their defaults are no use of the timer.

The optional PCB= parameter specifies the size of the interregion communication area. This area holds the PCBs included in the PSBs that are used by programs executing in this region. If you do not specify this parameter, the default value is 000. This allocates an area large enough to hold all the PCBs for the largest PSB defined to this IMS system and present in the active IMS.ACBLIBA/B. To specify a value, enter the number of 1024-byte blocks as a three-digit number (for example, 001).

Use the SSM parameter to reference a member in IMS.PROCLIB that identifies the DB2 subsystems that can be accessed from this BMP region. To prevent any access to a DB2 subsystem from the BMP, use the name of a null member (no entries). The default (a null value) allows the BMP region to attach to any of the subsystems declared to the IMS control region. If necessary, coordinate this parameter with the corresponding SSM parameter in the IMS procedure. Refer to the description in "System Control and Performance EXEC Parameters for the Control Region" on page 418.

The FMTO= parameter and the type of z/OS dump DD statements selected determine whether IMS dumps are formatted online or offline.

Use the APARM= parameter to specify execution-time parameters unique to this dependent region. This parameter specifies a character string for the application program or Data Capture exit routine. For batch or BMP regions, the APARM= can be used to specify the frequency for checkpoint calls. The APARM= can also be used to pass parameters to the Data Capture exit routine to indicate whether the data should be captured.

The maximum length of the parameter is 32 characters; the parameter must be enclosed in single quotes (') if special characters are used; embedded commas are not allowed. The INQY call with the ENVIRON subfunction is used to receive the APARM character string.

**Related Reading:** For more information, see *IMS Version 9: Application Programming: Database Manager*.

### Recovery-Related EXEC Parameters for a Batch Message Region

For a BMP program, you can use the CKPTID= parameter to set a restart position for the program processing. The IMSBATCH procedure generates CKPTID as a null parameter. To invoke restart, create a special version of the procedure containing the exact checkpoint identification. This can be a one- to eight-character extended checkpoint identifier generated by the program itself, or a twelve-character time-stamp identifier issued within a message from checkpoint processing. Alternatively, you can specify the value *LAST* as the identifier to cause the last recorded checkpoint to be used. One technique is to restart BMPs with JCL entered in the system reader and have the CKPTID symbolic parameter coded for the EXEC statement.

**Related Reading:** For more details regarding BMP restart for BMPs that use extended checkpoint, refer to *IMS Version 9: Operations Guide*.

In the case where the application program has a SPIE in effect, you specify whether the SPIE is to remain on or off during the DL/I call. Negated SPIEs are reinstated before returning to the application program. For performance reasons, it is recommended that the SPIE not be turned on and off for each DL/I call. With PL/I Release 5 and later releases, you can use the PL/I SPIE facility without having IMS reset the SPIE on each DL/I call.

### Security-Related EXEC Parameters for a Batch Message Region

The AGN parameter allows you to specify the application group name, which is associated with a set of transactions, PSBs, or LTERMs that are authorized to be used by this region. Declare these resources and the AGN in the Security Maintenance utility input to authorize them to be used. Include the same name to invoke IMS application group name security for this region. The IMSBATCH procedure generates the AGN as a null parameter. If other regions are being controlled through the use of AGNs, you risk permitting unauthorized access by this batch message program if you omit this parameter.

The parameter IMSID is related to security and to operations. You specify the one-to four-character identifier for the name of the operational IMS DB/DC environment. If the value does not match the current IMSID of any operating control region, the batch message region is not scheduled.

---

## Specifying IMS Buffer Pools

Three types of buffer pools control the IMS DL/I buffering services: the VSAM shared resource pool, the OSAM buffer pool, and the OSAM Sequential Buffer (SB) pool. This section describes the parameters that define the size and content of these pools. Each data set can have no more than one open ACB (VSAM access method control block).

VSAM constructs the VSAM shared resource pool based on parameters provided by the VSAM BLDVRP macro, which is issued during IMS initialization. This pool contains buffers to be used for VSAM data sets (both index and data components) and the input/output-related control blocks necessary to perform VSAM requests. The buffers are combined in subpools. All buffers within a subpool are of equal length.

The OSAM buffer pool is required for IMS online and batch operations. This pool contains buffers to be used for OSAM data set data components and the input/output-related control blocks necessary to perform OSAM requests. The buffers are combined in subpools. All buffers within a subpool are of equal length.

IMS dynamically constructs the OSAM Sequential Buffer (SB) pool when application programs or utilities using SB are active. To allow use of SB by online applications, you must provide a SBONLINE control statement in the DFSVSMxx member of the IMS.PROCLIB data set for an IMS DB/DC or DBCTL environment.

Each application program or utility using SB has its own SB buffer pools. You can control the use of SB by an application program or utility with PSBGEN, with control statements in the //DFSCTL data set, and with an SB Initialization exit routine.

**Related Reading:** See the explanation on “How to Request the Use of SB” in *IMS Version 9: Administration Guide: Database Manager*.

## VSAM Subpool Definition

Define VSAM subpools using the VSRBF statement, which is described in “Defining VSAM Subpools” on page 387. Specify the required subpools in the DFSVSAMP data set for batch environments or in the DFSVSMxx member of the IMS.PROCLIB data set for DB/DC environments. The VSAM buffers and control blocks are above the 16-megabyte line in most environments.

The minimum number of subpools is one and the maximum is eleven. The minimum IMS-calculated number of buffers in a subpool is three, and the maximum is 32767. This calculation is based on region type, number of PCBs, number of data set groups, and database organization.

A database is in sequential mode if **all** of the following conditions are true:

- The database operates in a batch type region (DLI or DBB)
- The database is referred to by only 1 PCB in the PSB
- The database organization is HISAM, single-segment HISAM, INDEX, or HIDAM accessed through unqualified GN calls

Buffer sizes can be .5K, 1K, 2K, 4K, 8K, and multiples of 4K up to 32K. IMS accepts any valid CI size up to 30K, but always uses a buffer size equal to or larger than the CI size. For example, a 30-K CI uses the 32-K subpool. If no VSAM files are used, the VSAM subpools need not be defined. A buffer handler pool is always built.



During DL/I database open, a data set is assigned a specific buffer subpool based on the CI size. The CI size must be equal to or less than the buffer size for the subpool assigned. The data and index components of a key-sequenced data set (KSDS) can be assigned to different subpools if their CI sizes are different and corresponding subpools exist. VSAM can assign a larger CI size to the INDEX component than to the DATA component when defining KSDS data sets that have small record sizes. For example, VSAM can assign a 512-byte DATA CI size and a 4096-byte INDEX CI size. You must be sure to define subpools that accommodate both INDEX and DATA components of a KSDS. (A listing of the VSAM catalog shows the CI sizes assigned to the components.) A single subpool can be defined with buffers large enough to contain the longest CI, or you can define several subpools that more nearly fit the different-sized CIs used by the programs.

**Related Reading:** Refer to *IMS Version 9: Administration Guide: System* for additional information on buffer pool structure and buffering techniques.

## OSAM Subpool Definition

OSAM subpools are defined using the IOBF and DBD statements as described in “Defining OSAM Subpools” on page 392. Specify the required subpools in the DFSVSAMP data set for batch environments or in the DFSVSMxx member of the IMS.PROCLIB data set for DB/DC environments. You can define multiple OSAM subpools as having the same buffer size (that is, the length parameters specified on the IOBF subpool definition statements are identical); then, you can direct given data sets to specific subpools. To take advantage of this flexibility, the desired subpool must have a user-defined ID (via the ID subparameter) and a DBD statement coded to identify the data set.

If you do not want to assign data sets to specific subpools, do not code the DBD subpool definition statement. If you define multiple OSAM subpools having the same buffer size and ID, IMS generates only one subpool for that buffer size. Within this subpool, the number of buffers is equal to the sum of the number of buffers specified for these subpools. The OSAM buffers and control blocks are above the 16-megabyte line. In a batch environment, the OSAM buffers and control blocks are located in the private address space. You must allocate a region size that is big enough to accommodate your OSAM buffer requirements.

## OSAM Buffer Pool Compatibility Definition

The OSAM buffer pool consists of one or more subpools defined through IOBF parameter statements. These statements are required in a DB/DC environment. Default member DFSVSM00 is used if the VSPEC parameter is not coded, but these statements are optional in a batch environment.

Omitting IOBF statements in a batch environment causes the OSAM subpools to be generated based on the following factors:

- The DMBs loaded for execution
- The values specified by the BUF= parameter on the EXEC statement

The BUF= parameter specifies the size of the OSAM buffer pool that is to be used in the subpool compatibility definition calculations. The size specified by the BUF= parameter does not necessarily define the actual size of the buffer pool. The specified size influences the eventual buffer pool size.

The buffer pool is generated as follows:

- IMS inspects each DMB loaded for execution for an OSAM DCB. The buffer length from the DCB is rounded up to the next appropriate subpool size (.5K, 1K, 2K, and so forth) and is used to create a potential subpool. DCBs with similar length requirements increment the occurrence count for that subpool.
- The potential number of buffers for each potential subpool is derived from the ratio of actual to least occurrences, multiplied by 4. The minimum number of buffers allowed per subpool is four.
- IMS allocates the actual number of subpool and buffers according to the previous ratio allocation using the buffer pool size specified by the BUF= parameter. The objective is to allocate at least four buffers per subpool without exceeding the specified size. The process begins with the subpool with the largest buffer size and continues to the subpool with the smallest. Depending on the specified buffer pool size, it might not be possible to generate all required subpools. IMS allocates only those subpools (largest to smallest), with a minimum of four buffers each, that fit in the specified size. A minimum of one subpool with a minimum of four buffers is allocated regardless of the buffer pool size defined. This subpool has a buffer size capable of handling the largest buffer size requirement.

**Example:** The BUF= parameter is specified as 10 KB. A scan of the loaded DMBs and the block size required by the OSAM DCBs indicates two subpools are required: one with 1-K buffers, and one with 4-K buffers. If each subpool has the minimum number of buffers, the total storage for the buffer pool is 20K. The size specified (by BUF=) is 10000 bytes. To support all requests, the subpool having 4-K buffers must be allocated. The subpool with 1-K buffers is not allocated. The buffer pool is allocated having one subpool with four 4-K buffers. The size is 16K (plus control areas), even though 10000 bytes are specified.

## Specifying VSAM and OSAM Subpools

The following example shows the control statements necessary to construct a single OSAM subpool with four 2K-buffers each and two VSAM subpools. The first subpool contains four 2048-byte buffers, and the second subpool contains fifteen 1024-byte buffers. The new KSDS records use mass insert.

```
IOBF=(2K,4,N,N)
VSRBF=2048,4
VSRBF=1024,15
OPTIONS,INSERT=SEQ
```

For information on specifying these control statements, see “Member DFSVSMxx” on page 383.

For IMS batch environments, the //DFSVSAMP data set provides the control statements listed in the example. Additionally, the //DFSVSAMP data set provides the GTF trace option. GTF trace records reflect the stream of I/O requests to the OSAM buffer handler.

**Related Reading:** For more information on GTF trace records, refer to *IMS Version 9: Administration Guide: System*. In an IMS environment, the DFSVSMxx member of the IMS.PROCLIB of the IMS system provides the statements. You should not place DFSVSAMP in the same PDS as your user application files. This causes an ABEND0C4 in DFSRTM00.

## Specifications for OSAM Sequential Buffering

The SBONLINE control statement allows use of OSAM Sequential Buffering (SB) in an IMS DB/DC or DBCTL environment. For information on specifying the SBONLINE control statement, see “Specifying Sequential Buffering for an Online System” on page 395.

One or more of the following factors controls the use of SB by a particular application program or utility:

- PSBGEN
- Control statements in the //DFSCTL data set
- An SB Initialization exit routine

**Related Reading:** For more information, see *IMS Version 9: Administration Guide: Database Manager*.

See also “Specifying Sequential Buffering Control Statements” on page 434 for more information on how to request use of SB through control statements in the //DFSCTL data set of the IMS batch region or IMS dependent online region.

---

## Making IMS and IMSRDR Procedures Accessible to z/OS

The IMS and IMSRDR procedures initiate the IMS control program region and message regions. These procedures are placed in IMS.PROCLIB by system definition and must be added to SYS1.PROCLIB.

**z/OS with JES2 or JES3:** Concatenate IMS.PROCLIB with SYS1.PROCLIB, or move the following procedures to SYS1.PROCLIB:

- IMS
- IMSRDR
- DBC
- DBRC address space procedure (as specified in DBRCNM= in the IMSCTRL macro)
- DL/I address space procedure (as specified in DLINM= in the IMSCTRL macro)
- DXRJPROC - IRLM procedure
- DFSMPR
- DFSWTnnn

The data set IMS.PROCLIB should be in the master catalog, otherwise, ensure that the volume serial number and unit are on the DD statement. If this is not done, JES2 is unable to initialize.

These procedures are described in Chapter 4, “Procedures,” on page 191.

Any dependent region jobs (IMSMSG, IMSWTnnn) to be specified in the IMS /START REGION command must be added to IMS.JOBS.

---

## Organizing PL/I Modules for Use with the PL/I Optimizer

You can decrease response time for those IMS applications that use the PL/I Optimizer by organizing the PL/I modules. Some organizational recommendations follow:

- Use several different program libraries, one for each region, putting only those modules required by the application in the library. Include in that library all supporting modules (such as the PL/I transient library modules).
- Concatenate the PL/I library into the message region STEPLIB.
- Put the required supporting modules in the link pack area. This is the recommended long-term solution for a virtual environment.
- Use IMS PRELOAD on the modules.

**Attention:** Do not use the PL/I Optimizing Compiler for multitasking during binding. Do not use SYS1.PLITASK as a SYSLIB data set.

---

## Coding COBOL Programs That Run under IMS

**Related Reading:** For information on recommended compiler options, bind attributes, and run-time options, see “Coding COBOL Programs that Run under IMS” in *VS Cobol II Application Programming Guide*. The LIBKEEP=YES run-time option is recommended. See also *Programming Language Considerations in an Information Management System (IMS) Environment*.

---

## Specifying Sequential Buffering Control Statements

This section describes the syntax and keywords of the six sequential buffering (SB) control statements: SBPARAM, SBIC, SBCO, SBSNAP, SBESNAP, and SNAPDEST.

**Related Reading:** For a more general description of SB concepts and terminology, refer to “Program Specification Block (PSB) definition” in *IMS Version 9: Utilities Reference: System* and *IMS Version 9: Administration Guide: Database Manager*.

Sequential buffering can be invoked by default for IMS utilities, such as online image copy, without requiring SB control statements, through the use of the SB Initialization exit routine.

**Related Reading:** For the description of the routine refer to “Sequential Buffering Initialization Routine” in *IMS Version 9: Customization Guide*.

SB control statements allow you to:

- Specify which I/O operations should be buffered by the SB buffer handler
- Override the default number of buffer sets to increase buffering performance
- Experiment with SB specifications without changing exit routines or regenerating the PSB
- Use SB problem determination aids

SB control statements are in a data set with the ddname of //DFSCCTL. You can provide a //DFSCCTL DD statement in the JCL of the IMS batch or IMS dependent online region. The //DFSCCTL DD statement must point to a sequential data set or a member of a partitioned data set. The record format of the //DFSCCTL data set must be F, FB, or FBS.

## Syntax of SB Control Statements

SB control statements are contained in 80-character records. Positions 72-80 are ignored.

Each SB control statement must begin on a new record. The first record must contain at least the statement name. IMS considers the first non-blank character the

start of the statement. The first word on the statement must be the statement name, such as SBPARM and SBIC, followed by one or more blanks.

Depending upon the type of SB control statement, one or more keyword parameters follow the statement name. Keywords must be followed immediately by an equal (=) sign and a value. If you specify multiple keywords, separate them with commas. If you specify multiple values, enclose them in parentheses. Individual keyword parameters must be completely contained on a single record.

A blank or a /\* signals the end of a statement. You can continue SB control statements on multiple records. Use a comma at the end of a line to indicate continuation.

An asterisk (\*) in the first position on a record indicates a comment. Elsewhere on a record, comments begin with /\*.

Two examples of proper syntax follow:

```
SBPARM ACTIV=COND,DB=SKILLDB,BUFSETS=6 /*Comment
```

```
SBPARM ACTIV=COND, /*This is a comment
          DB=SKILLDB, /*This is a comment
          BUFSETS=6 /*This is a comment
```

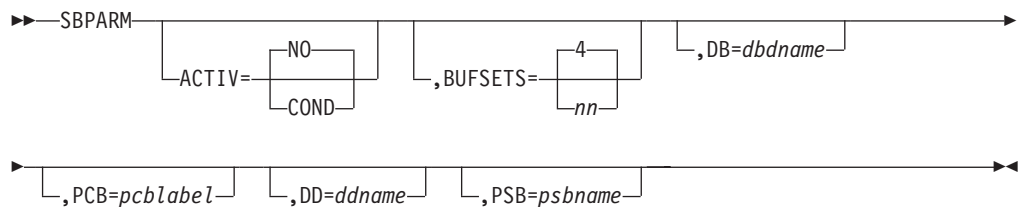
## SB Parameters (SBPARM) Control Statement

The SBPARM control statement is used to:

- Specify which I/O operations require sequential buffering (I/O operations can be specified by database, PCB, ddname, and PSB)
- Override SB default parameters

All SBPARM parameters are optional.

The format of the SBPARM control statement is:



Default values are underlined. The default value, however, might have been overridden by other specifications. If you provide the same SB option or parameter in more than one place, the following priority list applies (item 1 has the highest priority):

1. SB control statement specifications (the *n*th control statement overrides the *m*th control statement, where *n*>*m*)
2. PSB specifications
3. Defaults changed by the SB Initialization exit routine

If you specify multiple SBPARM control statements applying to the same I/O operation, the *n*th control statement overrides the *m*th one (where *n*>*m*). This lets you specify defaults on the first statement which can be overridden by subsequent statements.

The SBPARM control statement has two types of keyword parameters:

1. Option parameters, such as `ACTIV=` and `BUFSETS=`, which set and override an SB option or parameter, respectively
2. Qualification parameters, such as `DB`, `PCB`, `DD`, and `PSB`, which specify the I/O operations to which the SBPARM statement applies

### Option Parameters

Option parameters set or override an SB option or parameter. If an option parameter is not specified, the previously established value of the parameter is in effect. The previously established value might have been set by an IMS default, the SB Initialization exit routine, `PSBGEN`, or preceding SBPARM statements.

#### **ACTIV=**

Specifies if and when SB should be activated.

#### **NO**

Specifies that SB is not activated. `NO` is the default; however, the default might have been changed by what is specified in the SB Initialization exit routine, `PSBGEN`, or a preceding SBPARM control statement.

#### **COND**

Specifies that IMS monitors the I/O reference patterns of PCBs and eventually activates SB if analysis shows the I/O reference pattern is sequential, and the rate of I/O activity is high enough.

It is recommended that you specify `COND` for batch and BMP programs that sometimes or always process a database sequentially. SB is not recommended for short-running MPP, Fast Path, or CICS programs because of the overhead associated with initializing SB for each program execution.

If `ACTIV=` is not specified on an SBPARM control statement, IMS uses the previously established value of the `ACTIV` parameter. Therefore, omitting `ACTIV` lets you override the `BUFSETS` parameter without inadvertently changing SB `PSBGEN` specifications.

#### **BUFSETS=**

Specifies how many buffer sets are to be allocated for a given SB buffer pool. You can specify a value from 1 through 25. The default is 4. (The default can be changed by the SB Initialization exit routine.)

Each SB buffer pool consists of  $n$  buffer sets. Each buffer set has ten buffers. Each buffer is large enough to hold one OSAM data set block.

You must specify a value greater than 1 if SB is to use overlapped I/O.

A well-organized database might only require a `BUFSETS` value of 2 for efficient sequential buffering. Less well-organized databases, on the other hand, might require a `BUFSETS` value of 6 or more. One indicator of database organization can be found in the optional `//DFSSTAT` reports. If, for a sequential program, these reports show a low percentage of random I/O operations, this can indicate a well-organized database. Conversely, a high percentage of random I/O can indicate a poorly organized database. In either case, the `//DFSSTAT` reports can help you adjust the `BUFSETS` value to improve buffering performance.

### Qualification Parameters

Qualification parameters specify the I/O operations to which the SBPARM statement applies. You can associate SBPARM statements with a database, `PCB`, `ddname`, or `PSB`.

You can specify any combination of qualification parameters. If you specify more than one qualification parameter, the SBPARM statement applies only to I/O operations that satisfy all qualification parameters. If you do not specify any qualification parameters, the SBPARM statement applies to all I/O operations.

**DB=**

Specifies the DBD name (as specified in the PCB macro of PSBGEN) of those PCBs to which the SBPARM statement applies.

If multiple PCBs have the same DBD name, and you want different specifications for different PCBs, use the PCB= keyword. If the PCB refers to multiple database data sets and you want different specifications for these data sets, use the DD= keyword.

If no DB= keyword is provided, the SBPARM statement applies to all I/O operations that satisfy the other qualification keywords.

**PCB=**

Specifies the label coded on the database PCB during PSBGEN. Use this keyword to distinguish between multiple PCBs with the same DBD name.

If no PCB= keyword is provided, the SBPARM statement applies to all I/O operations that satisfy the other qualification keywords.

**DD=**

Specifies the ddname of the database data set to which the SBPARM statement applies. Use this keyword to distinguish between multiple data sets to which the PCB refers.

If no DD= keyword is provided, the SBPARM statement applies to all I/O operations that satisfy the other qualification keywords.

**PSB=**

Specifies the name of the PSB to which the SBPARM statement applies.

If no PSB=keyword is provided, the SBPARM statement applies to all I/O operations that satisfy the other qualification keywords.

Use of the PSB= keyword is strongly recommended for message regions. Normally, SB should not be activated for all programs running in the message region; rather, SB should be restricted to a very small number (if any) of these programs. If SB is to be used for more than one program, you need to provide one SBPARM statement for each program and code the PSBname of the program on the PSB= keyword.

PSB= is also useful if you decide to centralize all SB control statements in one common SB control statement file. In this case, PSB= identifies for each SBPARM statement, the PSB or program to which it applies.

**SBPARM examples**

Examples of the SBPARM control statement follow:

```
SBPARM ACTIV=COND                               /*Example 1
SBPARM ACTIV=COND,DB=SKILLDB                     /*Example 2
SBPARM ACTIV=COND,DB=SKILLDB,PCB=LABEL1         /*Example 3
SBPARM ACTIV=COND,DB=SKILLDB,BUFSETS=6         /*Example 4
SBPARM BUFSETS=6                                /*Example 5
SBPARM ACTIV=NO                                  /*Example 6
```

**Example 1:** SBPARM requests conditional activation of SB for all I/O operations. No default values for SB parameters are overridden.

**Example 2:** SBPARM requests conditional activation of SB for I/O operations associated with all PCBs coded with DBDNAME=SKILLDB during PSBGEN. No default values for SB parameters are overridden.

**Example 3:** SBPARM requests conditional activation of SB for all PCBs that satisfy both of the following conditions:

- DBDNAME=SKILLDB coded during PSBGEN
- The PCB label coded as LABEL1 during PSBGEN

**Example 4:** SBPARM requests conditional activation of SB for all PCBs coded with DBDNAME=SKILLDB during PSBGEN.

The BUFSETS parameter is set to 6 (instead of 4, which is the default). This increase in BUFSETS value can improve buffering efficiency if the database is poorly organized.

**Example 5:** SBPARM requests that the default values of BUFSETS be overridden. Increasing the value of BUFSETS to 6 (rather than using the default, which is 4) can improve buffering efficiency if the database is poorly organized. In this example, ACTIV has not been coded. As a result, the SBPARM statement does not change the previously established set of I/O operations being buffered by SB.

**Example 6:** SBPARM requests that SB not be used. The request applies to all I/O operations.

## SB Image Capture (SBIC) Control Statement

The SBIC control statement captures all internal calls to the SB buffer handler. The captured information is put in the X'5E' log record.

The X'5E' log records can be used as input to the SB test program (DFSSBHD0).

**Related Reading:** For detailed information on the SB test program, see *IMS Version 9: Utilities Reference: Database and Transaction Manager*.

The format of the SBIC control statement is as follows:

▶—SBIC—▶

## SB Compare Option (SBCO) Control Statement

The SBCO control statement asks the SB buffer handler to perform a self check to determine if SB is putting incorrect block images in the OSAM buffers, and if so, to provide problem determination information.

Problem determination information is provided in the form of a snap. In the IMS online environment, this snap is written to the IMS log. In the IMS batch environment, you can request that the snap be written to any of the following:

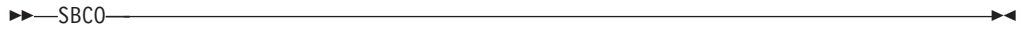
- The data set specified in the DFSDDLTO //PRINTDD DD statement
- A user-specified data set
- The IMS log

See “Snap Destination (SNAPDEST) Control Statement” on page 441 for a description of how to request these snap destinations. In the batch environment, the default snap destination is the data set specified in the //DFSSNAP DD statement.



**Related Reading:** For detailed information on the SBCO control statement, see *IMS Version 9: Diagnosis Guide and Reference*.

The format of the SBCO control statement is as follows:



### SB Snap (SBSNAP) Control Statement

The SBSNAP control statement snaps SB control blocks after each internal call of the OSAM buffer handler to the SB buffer handler. As an option, you can request that, for each internal SB call, the following additional information be snapped:

- The OSAM database block returned by the SB buffer handler to the OSAM buffer handler
- The SB buffers for the current data set group

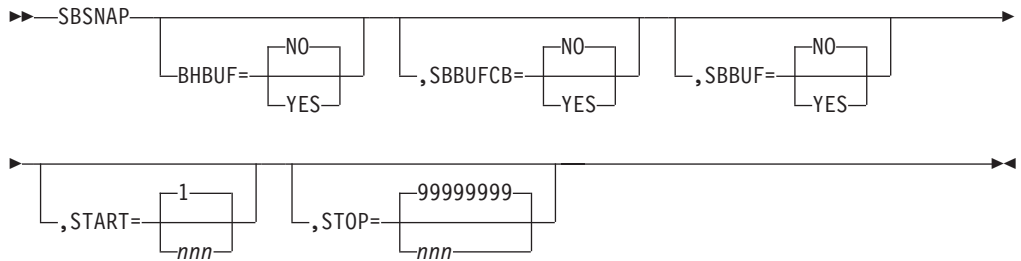
In IMS online environment, this snap is written to the IMS log. In the IMS batch environment, you can request that the snap be written to any of the following:

- The data set specified in the DFSDDLT0 //PRINTDD DD statement
- A user-specified data set
- The IMS log

See “Snap Destination (SNAPDEST) Control Statement” on page 441 for a description of how to request these snap destinations. In the batch environment, the default snap destination is the data set specified in the //DFSSNAP DD statement.

All SBSNAP parameters are optional.

The format of the SBSNAP control statement is as follows:



If you specify multiple SBSNAP control statements, the *n*-th control statement overrides the *m*th control statement (where *n*>*m*).

**Related Reading:** For additional information on the SBSNAP control statement, see *IMS Version 9: Diagnosis Guide and Reference*.

**BHBUFF=**

Specifies whether (YES) or not (NO) the OSAM buffer handler buffer is to be included in the snap produced at the end of each internal call to the SB buffer handler. The default is NO.

**SBBUFCB=**

Specifies whether (YES) or not (NO) the buffer control blocks of the SB buffer handler are to be included in the snap produced at the end of each internal call to the SB buffer handler. The default is NO.

**SBBUF=**

Specifies whether (YES) or not (NO) the buffers of the SB buffer handler are to be included in the snap produced at the end of each internal call to the SB buffer handler. The default is NO.

**START=**

Specifies that the snap is to start on the  $n$ th call to the SB buffer handler (instead of the first call). The default is 1.

The START= and the STOP= parameters are useful if you want to limit snapping to a specific period of time during which an application is running. The SBSNAP option often creates a large amount of output.

**STOP=**

Specifies that the snap is to be stopped on the  $m$ th call to the SB buffer handler (instead of the last call).

The default is that the snap is stopped after the last call.

**SB Evaluation Snap (SBESNAP) Control Statement**

The SBESNAP control statement snaps SB control blocks at the end of each periodic evaluation of the buffering process. The information in this snap can help you understand why IMS did or did not use SB.

In the IMS online environment, the snap is written to the IMS log. In the IMS batch environment, you can request that the snap be written to any of the following:

- The data set specified in the DFSDDLTO //PRINTDD DD statement
- A user-specified data set
- The IMS log

See “Snap Destination (SNAPDEST) Control Statement” on page 441 for a description of how to request these snap destinations. In the batch environment, the default snap destination is the data set specified in the //DFSSNAP DD statement.

All SBESNAP parameters are optional.

The format of the SBESNAP control statement is:



If you specify multiple SBESNAP control statements, the  $n$ th control statement overrides the  $m$ th control statement (where  $n > m$ ).

**Related Reading:** For additional information on the SBESNAP control statement, see *IMS Version 9: Diagnosis Guide and Reference*.

**EVAL=**

**NEG** requests a snap only when a periodical evaluation results in a decision not to use SB.

**POS** requests a snap only when a periodical evaluation results in a decision to use SB.

**ALL** requests a snap whenever a periodical evaluation of a data set group is made. The default is NEG.

## Snap Destination (SNAPDEST) Control Statement

The SNAPDEST control statement is applicable in batch environments only. SNAPDEST specifies the destination of the snap created by the SBSNAP, SBESNAP, or SBCO options. In the IMS online environment, SNAPDEST is ignored, and snap information is written to the IMS log.

The default destination for snap information in the batch environment is the data set specified in the //DFSSNAP DD statement.

If you specify an invalid destination in the SNAPDEST control statement, IMS tries to write the snap to the IMS log.

The format of the SNAPDEST control statement is as follows:



The destination specified in the SNAPDEST control statement can be any one of the following:

### DFSSNAP

Specifies that snap information is to be put in the data set associated with the //DFSSNAP DD statement.

### PRINTDD

Specifies that snap information is to be put in the data set associated with the //PRINTDD DD statement of the DL/I test program DFSDDLTO. This data set also contains information created by DFSDDLTO.

### ddname

Specifies that snap information is to be put in the data set with the specified ddname. The DD statement for this data set must conform to z/OS requirements for snap.

When the snap destination is ddname, DCB attributes are provided by IMS. You should not code them on the DD statement.

For this type of destination, each snap opens and closes the data set. If there are many snaps, this can result in significant overhead.

### LOG

Specifies that snap information is to be put in the IMS log.

When snaps are written to the log, the log record codes are as follows:

- X'67ED'- snap created by the SBESNAP option
- X'67EE'- snap created by the SBSNAP option
- X'67EF'- snap created by the SBCO option

You can select and print these log records using the File Select and Formatting Print utility (DFSERA10) and the Log Type X'67' Record Format and Print Module (DFSERA30).

**Related Reading:** Both of these programs are described in *IMS Version 9: Utilities Reference: System*.

---

## Specifying High-Speed Sequential Processing Control Statements

This section describes the syntax, keywords, and values of the two High-Speed Sequential Processing (HSSP) control statements: SETO and SETR.

HSSP control statements allow you to:

- Set up the environment in which you process a selected PCB with HSSP
- Create an image copy of a designated DEDB area, thereby reducing the amount of logged data
- Restrict access of a specific HSSP or non-HSSP application program to only designated DEDB areas

HSSO, HSSR, and HSSD control blocks are built from SETO and SETR statements. These control blocks—specifically those that represent image copy data sets and those that are used for UOW locking—are formatted for offline dumps. These control statements are in the DFSCCTL data set.

## Syntax of HSSP Control Statements

Each HSSP control statement must begin on a new line. IMS considers the first nonblank character the start of the statement. You must leave one or more blanks following the statement name. Keywords follow the statement name. Keywords must be followed immediately by an equal sign (=) and a value. If you specify multiple keywords, separate them with commas. If you specify multiple values, enclose them in parentheses.

A blank or a closing parenthesis signals the end of a statement. You can continue HSSP control statements on multiple records. Use a comma at the end of a line to indicate continuation.

An asterisk (\*) in the first position on a record indicates a comment. Elsewhere on a record, comments begin with a slash and asterisk (/ \*). You can include comments on each line with at least one blank between the keyword, value, or comma and the comment.

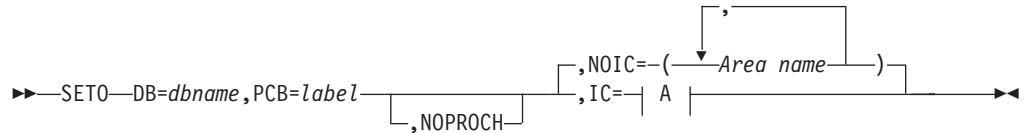
## Set Options (SETO) Control Statement

The SETO control statement allows you to specify the options in processing a PCB with HSSP. With SETO, you can:

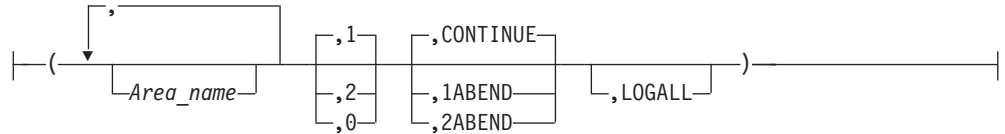
- Deactivate the HSSP option for a particular PCB
- Make an image copy of an updated area
- Specify the processing options in case of image copy failure

IMS interprets SETO when a region is started. After the region is active, you cannot alter the options.

The format of the SETO control statement is:



**A:**



**Keyword Parameters**

**DB=**

Specifies the name of the database to which SETO applies. Because HSSP is only applicable to DEDBs, only the names of these DEDBs are valid.

**PCB=**

Specifies a one- to eight- character label that identifies the PCB to which SETO applies. It must be identical to the label on the associated DEDB PCB. If you use the same PCB label more than once for the same database name, only the first SETO statement with that label is used.

**NOPROCH**

Specify NOPROCH if you want to deactivate the HSSP option that was initialized in PSBGEN (PROCOPT=H). If you specify this keyword, PROCOPT=P is set, and the IC option is ignored. You can only specify **one** HSSP PCB per database per PSB. If more than one HSSP PCB points to the same database, you should use the NOPROCH keyword to deactivate additional PCBs before the SETO statements are processed. If more than one HSSP PCB exists after the SETO statements have been processed, the region is not scheduled.

**NOIC=**

Specifies that no image copy should be made for all or some of the areas listed in DBDGEN. If you specify NOIC= without any area names, no image copy is made for any area in the database. You can use the option NOIC= with area names several times per SETO statement; however, you can specify a particular area only once. NOIC is the default if you do not specify IC=.

**IC=**

Specifies that an image copy should be made of all or some of the areas listed in DBDGEN. Allocation information about the image copy data sets is obtained from DBRC.

**Values:** The following values control the use of HSSP:

**Area name**

Lists the names of the areas to be image copied. If you need the areas to be processed in a specific order (other than the order in the DBDGEN), you must list them in the desired order and separate them with a comma. If the sequence of areas required is the same sequence as that in the DBDGEN, you only need to list the first and the last area names of the series and separate them with a hyphen. You can specify an area name only once. For example, A4,A1-A3,A5-A8 is valid, but A4,A1-A8 is invalid, because A4 is repeated.

If you specify IC= without any area names, an image copy of all areas in the database occurs. You can specify IC= without area names only once per SETO statement. You can use the option IC= with area names several times per SETO statement; however, you can specify an area only once.

Commas and parentheses are optional if you specify only one parameter on the IC= parameter. You must register the area in DBRC in order for image copying to take place.

**Related Reading:** See *IMS Version 9: Utilities Reference: Database and Transaction Manager* for more information.

- 1 Specifies that one image copy data set should be made. The default is one.
- 2 Specifies that two image copy data sets should be made. If you specify IC=2, two image copy data sets must be registered with DBRC for an area; otherwise, no image copy is done for that area.
- 0 Specifies that no image copy data sets should be made. IC=0 is the same as specifying NOIC.

#### **CONTINUE | 1ABEND | 2ABEND**

These keywords describe the action a program should take in the event of an image copy failure.

**CONTINUE** specifies that the program continues processing if the image copy option cannot be performed. Under this circumstance, messages are sent to the MTO console and the job log (WTP). CONTINUE can be abbreviated as C.

**1ABEND** specifies that if image copying cannot be completed for one data set, the program must abend. 1ABEND can be abbreviated as 1A.

**2ABEND** specifies that if image copying cannot be completed for two data sets, the program must abend. If you request IC=2 and one of the data sets is still usable, HSSP continues to write to it. 2ABEND can be abbreviated as 2A.

#### **LOGALL**

Specifies that X'5950' log records are to be logged for DEDB updates in an HSSP environment. If LOGALL is not specified, only the X'5947' log record is logged.

In an RSR environment, only X'5950' log records are logged, regardless of the specification of the LOGALL parameter.

If you specify only one parameter on the IC= keyword, you do not need to include the parentheses or the commas. Trailing commas for those parameters you do not include are also not required. For example, IC=1 is equivalent to IC=(,1,,) and IC=1,LOGALL is equivalent to IC=(,1,,LOGALL).

## **Set Range (SETR) Control Statement**

The SETR control statement specifies the processing range of PCBs to a database during scheduling of an application program. If a PSB has several PCBs pointing to the same database, you can restrict the access of each PCB to that database. Each program can only access data in the DEDB within the range defined.

The format of the SETR control statement is:

SETR—DB=dbname,PCB=label,AL=(Area name)

## Keyword Parameters

### DB=

Specifies the name of the database to which the SETR statement applies.

### PCB=

Specifies a one- to eight-character label that identifies the PCB to which the SETR statement applies. This label must be identical to the label on the associated DEDB PCB. If you use the same PCB label more than once for the same database name, only the first SETR statement with that label is used.

### AL=

Specifies the areas to which the PCB has access. The SETR area list does not have to match the SETO area list. (The SETR area list determines which areas can be processed. The SETO area list indicates which of the processed areas must have an image copy.) If you specify the SETR statement without a matching SETO statement (that is, the same DB= and PCB= values), the SETR statement processes the specified area list (AL=) values according to the PROCOPT option in PSBGEN.

## HSSP Control Statement Examples

The following are examples of SETO and SETR statements in the DFSCTL data set with their associated PCBs.

### Sample PSBGEN:

```

L1 PCB TYPE=DB,DBDNAME=DEDDB1,PROCOPT=A
L2 PCB TYPE=DB,DBDNAME=DEDDB2,PROCOPT=HI
L3 PCB TYPE=DB,DBDNAME=DEDDB3,PROCOPT=HRD
L4 PCB TYPE=DB,DBDNAME=DEDDB4,PROCOPT=HA
L5 PCB TYPE=DB,DBDNAME=DEDDB5,PROCOPT=HG
  
```

### Sample DFSCTL set option statements:

```

SETO DB=DEDDB1,PCB=L1,IC=1           /* Example 1
SETO DB=DEDDB2,PCB=L2,IC=(1,CONTINUE) /* Example 2
SETO DB=DEDDB3,PCB=L3,IC=(A1,A3-A6,2,1A) /* Example 3
SETR DB=DEDDB3,PCB=L3,AL=(A1-A5)
SETO DB=DEDDB4,PCB=L4,IC=(2,2ABEND)   /* Example 4
SETO DB=DEDDB5,PCB=L5,NOIC=(A4-A5)    /* Example 5
SETO DB=DEDDB5,PCB=L5,IC=1,LOGALL
  
```

### Example 1

This SETO statement matches the PCB with label L1. However, the PROCOPT=A defines it as a non-HSSP PCB. The SETO statement is ignored.

### Example 2

This SETO statement matches the PCB with label L2. The PROCOPT=HI defines it as an HSSP PCB.

**IC=** Indicates a list of values for the image copy option.

- 1** Requests one image copy of all referenced areas. A referenced area is an area for which a DL/I call has been issued.
- CONTINUE** Indicates that processing should continue if image copying cannot be performed.

**Example 3**

This SETO statement matches the PCB with label L3.

- A1,A3–A6** Requests an image copy of each indicated area when referenced by the application.
- 2** Requests two image copies of all referenced areas.
- 1ABEND** Requests that the program must abend if one image copy fails.

This SETR statement matches the PCB with label L3. The AL=(A1=A5) sets the range of areas to which PCB L3 has access.

**Example 4**

This SETO statement matches the PCB with label L4.

- 2** Requests two image copies of all referenced areas
- 2ABEND** Requests that the program must abend if both image copies fail. If one image copy data set is usable, the program continues to write on it.

**Example 5**

This SETO statement matches the PCB with label L5.

- NOIC=** Excludes the areas for which HSSP image copying is to be done.
- A4–A5** Indicates the areas for which no image copies are requested.

**Example 6**

This SETO statement matches the PCB with label L5 The PROCOPT=HI defines it as an HSSP PCB.

- IC=** Indicates a list of values for the image copy option.
- 1** Requests one image copy of all referenced areas. A referenced area is an area for which a DL/I call has been issued.
- LOGALL** Indicates that updates are logged in X'5950' log records.

---

## Supporting CCTL Users with DBCTL Databases

DBCTL is an IMS facility that allows access to DL/I full-function databases, HALDB, and data entry databases (DEDBs) from a subsystem called a coordinator controller subsystem (CCTL). A CCTL is the transaction management subsystem that communicates with DBCTL using the database resource adapter (DRA).

The following sections describe the installation tasks required of the CCTL to enable the DRA.



**Related Reading:** CCTL programming requirements for using the DRA interface appear in the *IMS Version 9: Administration Guide: Database Manager*. Specific installation concerns related to CICS as a CCTL are covered in Chapter 10, “Accessing IMS Databases with CICS,” on page 471.

Figure 34 illustrates the relationship of the DRA interface, CCTL and DBCTL.



Figure 34. DRA Interface to DBCTL

As part of the DBCTL definition process, you should consider CCTL requirements such as database, PSB, or buffer pools. For sample input for DBCTL stage 1 definition, see “IMS DBCTL Environment” on page 521.

## Preparing a CCTL

Two steps you must take to prepare a CCTL for enabling a DRA environment are:

1. The DRA startup/router routine (DFSPRRC0) must be in a CCTL load library. The routine can be copied from the IMS.SDFSRESL which is built by the IMS definition process. Or, the IMS.SDFSRESL can be concatenated in the CCTL step library.
2. The DFSPZPxx load module must be in a CCTL load library; it is the DRA startup table. The xx is the startup table name suffix that the CCTL specifies on the DRA initialization request. The source code for DFSPZPxx is DFSPZP00 in the IMS distribution library, DLIB. After all modifications have been made to DFSPZPxx, it is assembled using IMS DLIBS. A default load module, DFSPZP00, is in the IMS.SDFSRESL. It has default values for all but two of the required DRA initialization parameters, all of which can be overridden on the initialization (INIT) request itself.

The rest of the DRA code must reside in a load library that is dynamically allocated by DFSPRRC0. The DDNAME and DSNAME of this load library are specified in the startup table or initialization request. The default DSNAME is IMS.SDFSRESL because all DRA code resides here throughout the IMS definition process.

## The DRA Startup Table

This section presents sample source code for DFSPZP00 and describes the parameters specified through the DFSPRP macro in DFSPZP00.

### Sample DFSPZP00 Source Code

The DRA startup table, DFSPZPxx, is created by assembling the DFSPZPxx module. The following code is the actual source code for DFSPZP00. To define other variations of DFSPZPxx, modify this code through the DFSPRP macro. Specify the DRA parameters as keywords on the DFSPRP macro. The keywords and their descriptions appear following this sample:

```

      TITLE 'DATABASE RESOURCE ADAPTER STARTUP PARAMETER TABLE'
DFSPZP00 CSECT
*****
*                                                                 *
*  MODULE NAME: DFSPZP00                                         *
*                                                                 *
*  DESCRIPTIVE NAME: DATABASE RESOURCE ADAPTER (DRA)            *
*          STARTUP PARAMETER TABLE.                             *
*                                                                 *

```

```

*
* FUNCTION: TO PROVIDE THE VARIOUS DEFINITIONAL PARAMETERS
*           FOR THE COORDINATOR CONTROL REGION. THIS
*           MODULE MAY BE ASSEMBLE BY A USER SPECIFYING
*           THEIR PARTICULAR NAMES, ETC. AND LINKEDITED
*           INTO THE USER EXECUTION LIBRARY AS DFSPZPXX.
*           WHERE XX IS EITHER 00 FOR THE DEFAULT, OR ANY
*           OTHER ALPHANUMERIC CHARACTERS.
*
* KEYWORDS FOR THE DFSPRP MACRO:
*   DSECT=NO-A DSECT STATEMENT FOR PRP WILL NOT BE
*             GENERATED (LABEL DFSPRP WILL BE ON DS 00).
*
*   FUNCLV=  DEFAULT (1). ADAPTER FUNCTIONAL LEVEL.
*
*   DDNAME=  1 TO 8 CHARACTER DD NAME TO BE USED WITH
*             DYNAMIC ALLOCATION OF THE DBCTL EXECUTION
*             LIBRARY.
*             DEFAULT (CCTLDD).
*
*   DSNAME=  1 TO 44 CHARACTER DATASET NAME OF THE
*             DBCTL EXECUTION LIBRARY.
*             DEFAULT (IMS.SDFSRESL).
*
*   DBCTLID=XXXX-NAME OF THE DBCTL REGION
*             DEFAULT = SYS1
*
*   USERID=XXXXXXXX-NAME OF THE USER REGION
*
*   MINTHRD=XXX-MINIMUM NUMBER OF THREADS TO BE
*             AVAILABLE (MAXIMUM NUMBER IS 255)
*             DEFAULT = 1
*
*   MAXTHRD=XXX-MAXIMUM NUMBER OF THREADS TO BE
*             AVAILABLE (MAXIMUM NUMBER IS 255)
*             DEFAULT = 1
*
*   TIMER=XX-IDENTIFY TIMER VALUE IN SECONDS (DEFAULT 60)
*
*   FPBUF=XXX-NUMBER OF FAST PATH BUFFERS TO BE ALLOCATED
*             AND FIXED PER THREAD (DEFAULT 00)
*
*   FPBOF=XXX-NUMBER OF FAST PATH OVERFLOW BUFFERS TO BE
*             ALLOCATED PER THREAD (DEFAULT 00)
*
*   SOD=X-OUTPUT CLASS TO BE USED FOR SNAP DUMP OF
*             ABNORMAL THREAD TERMINATIONS (DEFAULT A)
*
*   TIMEOUT=XXX-DRATERM TIMEOUT VALUE IN SECONDS(DEFULT 60)*
*
*   CNBA=XXXX TOTAL FP NBA BUFFERS FOR CCTL
*
*   AGN=XXXXXXXX-1 to 8 CHARACTER APPLICATION GROUP NAME
*
* LOCATION: PRIVATE STORAGE, USER KEY
*
* THIS MODULE CONTAINS NO EXECUTABLE CODE.
*
*****
EJECT
DFSPRP DSECT=NO
END

```

### DFSPRP Macro Keywords

The following information describes DRA table parameters specified through the DFSPRP macro in DFSPZP00.

**AGN=** Is a one- to eight-character application group name that is used as part of the DBCTL security function.

**Related Reading:** See *IMS Version 9: Administration Guide: System* for more information.

**CNBA=**

Is the total number of Fast Path NBA buffers for the CCTL's use.

**Related Reading:** For a description of Fast Path DEDB buffer use, see *IMS Version 9: Administration Guide: System*.

**DBCTLID=**

Is the four-character name of the DBCTL region. This is the same as the IMSID parameter in the DBC procedure. The default name is SYS1. For more information on the DBC procedure, see "DBC Procedure" on page 198.

**DDNAME=**

Is a one- to eight-character ddname to be used with the dynamic allocation of the DBCTL execution library. The default ddname is CCTLDD. This library must contain the DRA modules.

**DSNAME=**

Is a one- to forty-four-character data set name of the DBCTL execution library. This library must contain the DRA modules and must be z/OS authorized. The default dsname is IMS.SDFSRESL.

**FPBOF=**

Is the number of Fast Path DEDB overflow buffers to be allocated per thread. The default is 00.

**Related Reading:** For a description of Fast Path DEDB buffer use, see *IMS Version 9: Administration Guide: System*.

**FPBUF=**

Is the number of Fast Path DEDB buffers to be allocated and fixed per thread. The default is 00.

**Related Reading:** For a description of Fast Path DEDB buffer use, see *IMS Version 9: Administration Guide: System*.

**FUNCLV=**

Specifies what level of the DRA the CCTL supports. The default is 1. There is no other value possible.

**MAXTHRD=**

Is the maximum number of DRA thread TCBs to be available at any given time. The maximum number is 255. The default is 1.

**MINTHRD=**

Is the minimum number of DRA thread TCBs to be available at any given time. The maximum number is 255. The default is 1.

**SOD=** Is the output class to be used for a SNAP DUMP of abnormal thread terminations. The default is A.

**TIMEOUT=**

Is the amount of time (in seconds) a CCTL should wait for the successful completion of a DRA TERM request. This value should be specified only if the CCTL is coded to use it. This value is returned to the CCTL upon completion of an INIT request. The default is 60 seconds.

**TIMER=**

Is the amount of time (in seconds) between attempts of the DRA to identify itself to DBCTL during an INIT request. The default is 60 seconds.

**USERID=**

Is an eight-character name of the CCTL region.

---

## Managing an IMSplex Environment with a Common Service Layer

An IMSplex is a group of IMS address spaces (control, manager, or server) that work together as a unit. Typically, but not always, the address spaces:

- Share either databases or resources or message queues (or any combination)
- Run in an sysplex environment
- Include an IMS Common Service Layer

Systems management becomes more complex as multiple IMS systems work together in the sysplex environment. The Common Service Layer (CSL) provides the infrastructure for systems management needed for IMSplex features through three IMS address spaces, called IMS CSL managers:

**Operations Manager (OM)**

Provides an application programming interface that allows commands to be entered from a single point of control (SPOC) application to one or more IMSs and consolidates command responses that are to be returned to the application.

**Resource Manager (RM)**

Coordinates IMSplex-wide processes and maintains global resource information on a Coupling Facility list structure called a resource structure.

**Structured Call Interface (SCI)**

Provides a common interface for communications between IMSplex members.

A single CSL can service multiple IMSplex members.

**Related Reading:** For information about defining and configuring a CSL, defining the CSL components, and other details about the CSL components, see the *IMS Version 9: Common Service Layer Guide and Reference*.

---

## Enabling ETO Support for ACF/VTAM Terminals

To use ETO, at least one user descriptor and one logon descriptor must exist. If at least one of each of these descriptors does not exist when you enable ETO, IMS abends with U0015 issuing message DFS3652.

**Related Reading:** For information on exit routines associated with ETO, refer to *IMS Version 9: Customization Guide*. For information on what is required to enable ETO in your installation, refer to the ETOFEAT= keyword in the “IMS Procedure” on page 231.

## ETO Descriptors

When you are enabling ETO support in your installation, you can optionally request that IMS produce ETO descriptors during system definition. You do this with the ETOFEAT keyword in the IMSCTRL macro (“IMSCTRL Macro” on page 87). Descriptors contain information about terminal devices and users that is required to establish dynamic sessions. Four types of descriptors are:

Logon

MFS Device  
 MSC  
 User

Descriptors that are created during system definition are placed in member DFSDSCMx, which resides in IMS.PROCLIB. If you perform subsequent system definitions of the same stage 1 input deck, the DFSDSCMx member are overwritten. If you use TSO or an z/OS utility to create descriptors, you can place the descriptors in member DFSDSCTx to avoid loss in the event that member DFSDSCMx is replaced. You can also use TSO or an z/OS utility to update descriptors in DFSDSCMx that were created during system definition.

**Descriptor Format**

The suffix for DFSDSCMx will always be the nucleus suffix. The default suffix for DFSDSCTy defaults to 0; however, you can specify it with the DSCT= JCL parameter. Each descriptor statement must be 80 characters in length. All statements are translated to upper case. The basic format of each of the descriptor types is shown in Table 50.

**Related Reading:** For more information on descriptor formats, refer to *IMS Version 9: Administration Guide: Transaction Manager*.

Table 50. Descriptor Type Formats

Column Number	Contains	Considerations
01	The descriptor type indicator:  L (logon) M (MSC) U (user) D (MFS device)	If an asterisk (*) is in column 01, the record is considered a comment record and is ignored.
03-10	The descriptor name	<ul style="list-style-type: none"> <li>All descriptor names must be 1-8 alphanumeric characters.</li> <li>The first character for logon and user descriptors must be alphabetic (A-Z, #, \$, or @).</li> <li>Duplicate descriptor names are allowed if they are grouped together, because more parameters might be required than can exist in an 80-byte record.</li> <li>For MSC descriptors, this column contains the link name.</li> </ul>
12-72	Parameters for each descriptor	<ul style="list-style-type: none"> <li>Parameters vary by type.</li> <li>They are delineated by blanks.</li> <li>They are typically in the same format as the equivalent parameters specified in the stage 1 input deck.</li> </ul>
73-80	Can contain sequence numbers	IMS ignores these columns.

**Rules for Creating Descriptors**

You should be aware of the following general guidelines when creating descriptor input streams.

**Keywords:**

- Separate one keyword or parameter set from another with one or more blanks.
- Do not include imbedded blanks within a keyword and its parameters.
- Separate a keyword from its parameters with an equal sign (=).
- Do not abbreviate keywords.

You can continue a keyword and its parameter set to the next statement if no intervening blanks appear at the end of the first statement or at the beginning of the parameters of the next statement. A continued statement still has the same descriptor type and name in columns 1-10; the continued specification begins in column 12. If you specify keywords, they must be accompanied by parameters. Keywords followed by blanks or commas are invalid.

**Parentheses:** Parentheses are optional if they follow a keyword that supports multiple parameters and only if the first parameter is specified. Otherwise, they are required.

**Parameters:**

- Separate multiple parameters for single keywords with commas.
- Specify at least one parameter for each keyword.

If the keyword has positional parameters, you can use commas when parameters are specified after a missing positional parameter. Following is an example of valid use:

```
LTERM=(ABC,,1)
```

This example is valid because LTERM has positional parameters.

The following example is invalid:

```
LTERM=(ABC,,1,)
OPTIONS=(TRANSRESP,)
OPTIONS=(,SYSINFO)
```

The LTERM example is invalid, because the comma must be followed by a must be a parameter. The OPTIONS examples are invalid, because the parameters for the OPTIONS keyword are not positional.

**Descriptor Overrides:** If a descriptor is defined in both DFSDSCTy and DFSDSCMx, the descriptor in member DFSDSCTy overrides the descriptor in member DFSDSCMx. No message is issued. If one descriptor is defined more than once in either member, error message DFS3661 is issued. IMS uses the last descriptor defined. If the same descriptor is defined once in member DFSDSCTy and one or more times in member DFSDSCMx, the descriptor in member DFSDSCTy is used, and no error message is issued.

**Restriction:** No more than 512000 records (excluding comments) are permitted for each individual descriptor.

## Logon Descriptors

When you request that ETO descriptors be created during system definition, a logon descriptor is created for each VTAM TYPE and TERMINAL macro set. Exceptions to this are:

- Any terminal defined as a primary or secondary MTO
- Any LUTYPE6 terminal defined as the XRF ISC link

ETO logon descriptors are not produced for these terminals.

Also produced during system definition is a default logon descriptor, which, reflects the TYPE and TERMINAL macro statements that define the unique characteristics of the largest number of terminals of that type. Terminals that cannot be represented by the default logon descriptor have unique logon descriptors created for them. You can keep and use these unique logon descriptors or discard them by:

- Operating these terminals based on the default logon descriptor
- Creating one or more of your own installation default logon descriptors and coding an installation exit routine DFSLGNX0, to select the required descriptor for logging on to a terminal

The system definition process generates up to 37 default logon descriptors for each device type. ETO descriptors are not generated during large system generation (LGEN). The suffix for each descriptor defaults to 0; the last character of the descriptor name ensures that the name is unique. This character is a blank for the most common descriptors, and then 0 through 9, and A through Z, for a total of 37 possible descriptors.

Terminal definitions that do not match the 37 most common terminal definitions generate comment statements with an asterisk in the first position and the rest of the statement is shifted over one position.

All user descriptors created during system definition are generated as comment statements to avoid getting error messages caused by a statically defined LTERM name.

Typically, one generic logon descriptor exists per unique LU type and option configuration. After the descriptor is created, the same descriptor can support any number of logons for the same LU type and option configuration.

## Logon Descriptor Format

The format of a logon descriptor is

```
L Descriptor name Parm(1) Parm(2) Parm(3)
L Descriptor name Parm(4) Parm(5) Parm(n)
```

The parameters that are required for a descriptor cannot fit into one record. You might need to use multiple records with the same descriptor name to define your descriptor.

**Related Reading:** Refer to *IMS Version 9: Administration Guide: Transaction Manager* for more information on descriptor formats.

## Logon Descriptor Parameters

Because logon descriptors are built from equivalent TYPE and TERMINAL macros, most definitions of logon descriptor parameters match the parameters found on those macro statements. Parameters that are not defined in this section are **not** supported for logon descriptors.

### descriptor type

Specifies that the descriptor is logon (L).

### descriptor name

Specifies the name of the logon descriptor name. Default names include DFS3270, DFS327P, DFSFIN, DFSSLU1, DFSSLU2, DFSSLUP, DFSLU61, and DFSNTO.

**ALOT=**

Specifies the auto logoff time in minutes. Valid values are 0 and from 10 to 1440. If the ALOT value is not specified, the value from the JCL member is used except for FINANCE, SLU P, and ISC. If ALOT is not specified on the logon descriptor or overridden by the logon exit (DFSLGNX0) for FINANCE, SLU P, and ISC, a value of 1440 is used (the value from the JCL member is ignored).

- ALOT= 0

The terminal is logged-off immediately when no signon is in effect. This specification is normally used in terminal sessions when the user is signed-on automatically during the logon process. During autologon, signon data can be provided in one of the following ways:

- Signon data supplied via the IMS /OPNDST command
- Signon data supplied by logon userdata (BIND)
- Signon data supplied by logon exit (DFSLGNX0)

There are two modes of operation for using ALOT=0, either of which can be set using the DFSINTX0 User Initialization Exit parameter list.

In default mode, when signon errors are encountered, the session is automatically signed off and then logged off; no message is sent. If you do not supply the DFSINTX0 exit, or you supply the exit and indicate default mode for ALOT=0, then signon data must be supplied during the logon process. All of the following error conditions result in automatic logoff:

1. A non-signon, or errors detected during signon or input processing, result in immediate logoff.
2. /SIGNOFF results in immediate logoff.
3. /SIGNON signs off the current user and signs on a new user. However, errors encountered during the signon process, such as detection of an incorrect or expired password, result in immediate logoff.

**Restriction:** Default mode should not be used for interactive terminal sessions that require a response to the DFS3649 message; these sessions will not wait for input signon and will logoff immediately.

In alternate mode, when signon errors are encountered, the session is automatically signed off, a message is sent, and the session is logged off. Signon data can be supplied but is not required. All of the following error conditions result in automatic logoff:

1. A non-signon error detected during input processing results in immediate logoff.
2. No signon data has been provided by the logon userdata (BIND) or the Logon Exit (DFSLGNX0).
3. A /SIGNOFF, or errors resulting from a /SIGNON, cause message DFS3649(A) (Signon Required) to be sent, and a fixed ten-minute timer set to wait for a new signon. If no signon occurs during that interval, then the session will be logged off.

- ALOT=(10-1439)

The session is terminated after the specified number of minutes have elapsed without a signed-on user.

- ALOT=1440

The session is never automatically terminated.

**ASOT=**

Specifies the auto signoff time, in minutes. Valid values are 0 and from 10 to



1440. You can override this value during signon with the user descriptor. If ASOT is not specified on either the user descriptor, the logon descriptor, or overridden by either the logon (DFSLGNX0) or signon (DFSSGNX0) exits, the value from the JCL member is used except for FINANCE, SLU P, and ISC. If ASOT is not specified on either the user or logon descriptor for FINANCE, SLU P, and ISC, a value of 1440 is used (the value from the JCL member is ignored).

- ASOT=0

The user is signed off immediately when no output is available to be sent. ASOT=0 is normally specified when no IMS input or output is available, or after the last available output message completes.

**Restriction:** Do not specify ASOT=0 for interactive terminals, for example 3270s or SLU2.

- ASOT=(10-1439)

The user is signed off after the specified number of minutes have elapsed without terminal activity, irrespective of message status.

- ASOT=1440

The user is never automatically signed off.

#### **BACKUP=**

Specifies the control of automatic restart (BTAM) or session switching (VTAM) after takeover. Use only when HSB=YES on the IMSCTRL macro.

This keyword is the same as the BACKUP= keyword in the TYPE macro. See "TYPE Macro" on page 188 for more information on keyword values.

#### **COMPTn=**

Where *n* is 1-4, specifies the first through fourth components of this logon descriptor. This keyword is only valid for the following device types: 3270, FINANCE, SLU 1, SLU P, and LU6.1 (ISC). For UNITYPE=FINANCE, these four keywords are the same as the four operands on the TERMINAL macro's COMPT keyword. The parameters for this keyword are positional. See "TERMINAL Macro" on page 143 for more information on this keyword for other device types.

COMPT=PTR1 for UNITYPE=3270, UNIT=3275 should be specified as COMPT2=PTR1.

#### **EDIT=**

Specifies the name of a user-supplied physical terminal output and input edit routine for the terminals in this communication description set.

This keyword is the same as the EDIT= keyword in the TYPE macro. See "TYPE Macro" on page 188 for more information on keyword values.

#### **FEAT=**

The parameters for this keyword are not positional.

See the description of the FEAT= keyword in "TERMINAL Macro" on page 143 and "TYPE Macro" on page 188 for more information on this keyword.

#### **OPTIONS=**

Specifies certain communication options associated with this terminal. Parameters for this keyword are not positional.

The ARNR|NRNR option specifies whether (ARNR) or not (NRNR) IMS should enable Rapid Network Reconnect (RNR) support for ACF/VTAM persistent sessions as defined on the ACF/VTAM APPL statement.

Aside from the exceptions noted in this section, the communications options described in the `OPTIONS=` keyword of the `TYPE` and `TERMINAL` macros are supported for logon descriptors. See “`TERMINAL Macro`” on page 143 and “`TYPE Macro`” on page 188 for more information on this keyword.

The following options are not supported for logon descriptors:

- `NOSHARE|SHARE`
- `ACK|OPTACK`
- `NORESP|FORCRESP|TRANRESP`

`NORESP|FORCRESP|TRANRESP` can be specified on the user descriptor.

The `COPY|NOCOPY` option is supported for logon descriptors for dynamic SLU 2 terminals, but not for dynamic non-SNA VTAM 3270s.

#### **OUTBUF=**

Specifies the size of the IMS output buffer to be used for workstations.

This keyword is the same as the `OUTBUF=` keyword on the `TERMINAL` macro. See “`TERMINAL Macro`” on page 143 for more information on this keyword.

#### **PTRSIZE=**

Specifies the number of print positions of the 3284 or 3286 printer. The default is 120.

If `PTRSIZE=IGNORE` is specified with `FEAT=IGNORE` in its `DEV` statement, MFS always uses the device output format (DOF) when editing output for this device. The existing parameters and control block values, including default parameters, are unchanged.

#### **SEGSIZE=**

Specifies segment size. Acceptable values are from 256 to 32000 bytes. The default is 256. For information on calculating segment size, see the description of the `SEGSIZE=` keyword in “`TERMINAL Macro`” on page 143.

#### **UNIT=**

Specifies the terminal for the previously defined line group.

This keyword is the same as the `UNIT=` keyword on the `TERMINAL` macro. See “`TERMINAL Macro`” on page 143 for more information on this keyword.

#### **UNITYPE=**

Specifies the `UNITYPE` of the terminal. Valid `UNITYPE`s include 3270, `FINANCE`, `SLUTYPE1`, `SLUTYPE2`, `SLUTYPEP`, `LUTYPE6`, or `NTO`. 3601 is an invalid `UNITYPE`. If not specified, a default `UNITYPE` of `SLUTYPE2` is used.

### **Logon Descriptor Example**

Following is an example of a logon descriptor:

```
L MKT01LU2 UNITYPE=SLUTYPE2 ALOT=30
```

## **MFS Device Descriptors**

MFS device descriptors are used by the MFS Device Characteristics Table (DCT) utility. This utility updates screen information, for example 3270 screen sizes and feature information, in the DCT and generates new MFS default formats without system definition.

You use MFS device descriptors to define characteristics of terminals added dynamically that have different characteristics from statically defined terminals.

**Related Reading:** See *IMS Version 9: Utilities Reference: Database and Transaction Manager* for more information about how to use the DCT utility.

### MFS Device Descriptor Format

The format of an MFS device descriptor is

```
D Descriptor name Parm(1) Parm(2) Parm(3)
```

### MFS Device Descriptor Parameters

The three parameters supported for MFS device descriptors are TYPE=, SIZE=, and FEAT=.

#### **Example of MFS Device Descriptors:**

```
Co1      Co1
1        12
D        TYPE=3270-A04 SIZE=(43,80) FEAT=IGNORE
```

See the parameter descriptions in “TERMINAL Macro” on page 143 for more information.

## MSC Descriptors

MSC descriptors relate remote NAME macros to MSC links defined during system definition. When you specify with ETOFEAT that descriptors are to be created during system definition, an MSC descriptor is created for each MSNAME macro statement.

### MSC Descriptor Format

The format of an MSC descriptor is

```
M Link name name1 name2 namen
```

An MSC descriptor contains a link name rather than a descriptor name. The link name is the name of the retrieved descriptor. The name parameters represent any valid remote LTERM name that can be accessed through a link name. The parameters that are required for a link name cannot fit into one record. You might need to use multiple records with the same link name to define your descriptor.

### MSC Descriptor Parameters

The parameters that are supported for MSC descriptors are as follows:

#### **descriptor type**

Specifies that the descriptor is MSC (M).

#### **descriptor name**

Specifies the linkname from the MSNAME macro.

#### **remote LTERM name**

Specifies the name of a logical terminal associated with a physical terminal defined in a remote IMS system.

### MSC Descriptor Examples

Following are examples of MSC descriptors:

```
M REMSYS01 REM01AAA REM01BBB REM01CCC REM01DDD REM01EEE REM01FFF
M REMSYS01 REM01GGG REM01HHH REM01III
```

## User Descriptors

User descriptors are another form of descriptor created during system definition for ETO support. Three types of user descriptors are:

- DFSUSER

- Node
- Installation-created

The node user descriptors are essentially migration aids, allowing message queues and options associated with a particular terminal to remain unchanged during migration. They do not, however, provide output security.

**Related Reading:** For information on the format of user descriptors, refer to *IMS Version 9: Administration Guide: Transaction Manager*.

User descriptors are generated from each VTAM TERMINAL or VTAMPOOL SUBPOOL macro statement. Those descriptors created from VTAM TERMINAL macro statements, such as node user descriptors, have the same name as the terminal. Descriptors created from VTAMPOOL SUBPOOL macro statements have the same name as the subpool.

For user descriptors created during IMS system definition from the SUBPOOL macro statement, the response option (TRANSRESP, NORESP, or FORCRESP) cannot be set. This is because the response option is defined on the TERMINAL statement for static definitions. You need to add the appropriate response option for your installation to any user descriptor created from a SUBPOOL statement.

### DFSUSER Descriptor

When you request that IMS build ETO descriptors, IMS creates a single default user descriptor, DFSUSER, which defines default user characteristics. The DFSUSER descriptor represents the most common set of user options. With this descriptor, IMS can dynamically create a user and message queue structure for a signon request when no other user descriptor is available. The message queue name is the same as the user ID. You can also use DFSUSER to add queue names and other options through the DFSSGNX0 Signon exit routine.

**Related Reading:** Refer to *IMS Version 9: Customization Guide* for more information on how to use the DFSSGNX0 exit routine.

It is recommended that you use the DFSUSER descriptor for most of your users after your migration to ETO is completed.

### Node User Descriptors

Node user descriptors are created during system definition. One is generated for each terminal in the IMS system definition. Node user descriptors are created even if they match DFSUSER options, except for ISC terminals defined for parallel-session support.

### Installation-Created Descriptors

Installation-created user descriptors are designed by your installation to meet your criteria. The name of the installation-created user descriptor is the same as the user ID.

### User Descriptor Format

The format for a user descriptor, regardless of whether it is created by installation, node, or DFSUSER, is:

```
U username parm1 parm2 parmn
```

U is the descriptor type (USER), and Username is DFSUSER, the user ID, or the node user name from the name field in the TERMINAL or SUBPOOL macro statement supplied by the installation.

parm (1..n) can be any of the following parameters: ASOT=, LTERM=, OPTIONS=, AUTLGN=, AUTLMOD=, AUTLDESC=, AUTLID=

### User Descriptor Parameters

Unless specifically noted in this section, the definition of a given user descriptor parameter is the same as that found for the equivalent NAME ("NAME Macro" on page 129) and SUBPOOL macro statement ("SUBPOOL Macro" on page 141). Also, unless specifically noted, the general rules for creating descriptors apply to the following user descriptors. Parameters not defined in this section are not supported for user descriptors.

The following parameters are valid on the user descriptor:

#### ASOT=a

*a* is 0 or a value from 10 to 1440. The default comes first from the logon descriptor parameter ASOT. If ASOT is not specified on either the user or logon descriptor, the value from the JCL member is used except for FINANCE, SLU P and ISC. If ASOT is not specified on either the user or logon descriptor for FINANCE, SLU P, and ISC, or overridden by the logon (DFSLGNX0) or (DFSSGNX0) exits, then a value of 1440 is used, the value from the JCL member is ignored.

#### AUTLDESC=d

*d* is a one- to eight-byte alphanumeric logon descriptor name, the first character of which is alphabetic (A-Z, \$, #, @). AUTLDESC=d defines the characteristics of the terminal to be autologged on. This parameter is ignored for LU 6.1 devices.

#### AUTLGN=b

*b* is a one- to eight-byte alphanumeric LU name for auto logon, the first character of which is alphabetic (A-Z, \$, #, @). Autologon allows IMS to logon and signon your terminal automatically. If you specify the autologon option for a user, the queueing of data to any of the user queues causes IMS to establish a session.

**Related Reading:** For more information, see *IMS Version 9: Administration Guide: Transaction Manager*.

#### AUTLID=g

*g* is a one- to eight-byte alphanumeric ISC other system half session qualifier, of which the first character is alphabetic (A-Z, \$, #, @). If the other system is IMS, then this is the name of an ISC user in that system. The AUTLGN parameter must be used with the AUTLID parameter when requesting autologon for an ISC parallel session.

#### AUTLMOD=e

*e* is a one- to eight-byte alphanumeric mode table for the autologon terminal.

#### LTERM=(f,h,i,j)

*f*, *h*, *i*, and *j* are all positional parameters. *f* is a one- to eight-byte alphanumeric LTERM name. *h* is either ULC or UC; the default is ULC. *i* is the COMPT parameter, and *j* is the ICOMPT parameter. COMPT and ICOMPT are the same as defined by keyword parameters on the NAME macro. If *f* is the only parameter supplied, the parentheses can be omitted.

A maximum of eight queues can be specified for each user descriptor.

If no LTERM keyword is specified, the user descriptor control block structure is built with one queue. The name used as the descriptor name is the default.

The queue names specified on the LTERM parameter keyword for the user descriptors must be unique. That is, two user descriptors cannot be created with the same queue name. If duplicate LTERM names are specified, message DFS3669 is issued.

If the LTERM keyword is specified on the user descriptor, the queue name must also be specified. The DFSUSER descriptor must also follow this convention (for example, with queue name DFSUSER).

The queue names (and Remote LTERM names) must follow the same naming conventions as the user names because they can, in some circumstances, take on the same name as the user name. The user name has strict naming conventions because it must follow RACF naming conventions. Therefore, these names must be alphanumeric (A-Z, 0-9, \$, #, @), but the first character in the name must not be 0-9.

**OPTIONS=j,k**

*j* is a response mode of FORCRESP, TRANRESP, or NORESP. The default response mode is dependent upon the device type. If the response mode is not explicitly set in the user descriptor, the default cannot be set until signon, when the device type is known. The default setting for each device type, during signon, is indicated in Table 51.

Table 51. Default Response Mode Values

Device Type	Default Response Mode
3270	NORESP
FINANCE	NORESP
LU 6	TRANRESP
NTO	TRANRESP
SLU 1	TRANRESPX
SLU 2	NORESP
SLU P	NORESP

*k* is a MSGDEL option of SYSINFO, NONIOPCB, or NOTERM. The default is SYSINFO.

If OPTIONS=NOTERM is specified for a user descriptor, when that user descriptor is used for a sign on of a FINANCE, SLU P, or ISC terminal, SYSINFO issued instead of NOTERM.

The OPTION parameters are not positional. Refer to the MSGDEL and OPTIONS parameters of the TERMINAL macro for descriptions.

**RCVYCONV=**

Specifies whether the status of a conversation can be recovered (Y) or not (N). RCVYCONV applies to conversation status, not to output messages. Even if the conversation status is not recovered, conversation output continues to be recoverable and is delivered asynchronously.

Table 52 describes the default values for RCVYCONV as they relate to the values specified on the SRMDEF keyword.

Table 52. RCVYCONV Values Related to SRMDEF Values

	RCVYCONV=YES	RCVYCONV=NO
SRMDEF=GLOBAL	Valid (default)	Valid
SRMDEF=LOCAL	Valid (default)	Valid

Table 52. RCVYCONV Values Related to SRMDEF Values (continued)

	RCVYCONV=YES	RCVYCONV=NO
SRMDEF=NONE	Invalid	Valid (default)

If RCVYCONV is specified incorrectly, message DFS1920I is issued and IMS uses the appropriate default from Table 52 on page 460.

**RCVYFP=**

Specifies whether the status of Fast Path can be recovered (YES) or not (NO). RCVYFP applies to conversation status and Fast Path output.

Table 53 describes the default values for RCVYFP as they relate to the values specified on the SRMDEF keyword.

Table 53. RCVYFP Values Related to SRMDEF Values

	RCVYFP=YES	RCVYFP=NO
SRMDEF=GLOBAL	Valid (default)	Valid
SRMDEF=LOCAL	Valid (default)	Valid
SRMDEF=NONE	Invalid	Valid (default)

**Restriction:** If SRMDEF=LOCAL, STSN recoverability (RCVYSTSN) and fast path recoverability (RCVYFP) must be the same. They must both specify YES or both specify NO.

If RCVYFP is specified incorrectly, message DFS1920I is issued and IMS will use the appropriate default from Table 53.

**RCVYSTSN=**

Specifies whether the status of STSN terminals (SLUP, FINANCE, and ISC) can be recovered (Y) or not (N).

Table 54 describes the default values for RCVYSTSN as they relate to the values specified on the SRMDEF keyword.

Table 54. RCVYSTSN Values Related to SRMDEF Values

	RCVYFP=YES	RCVYFP=NO
SRMDEF=GLOBAL	Valid (default)	Valid
SRMDEF=LOCAL	Valid (default)	Valid
SRMDEF=NONE	Invalid	Valid (default)

**Restrictions:** If SRMDEF=LOCAL, STSN recoverability (RCVYSTSN) and fast path recoverability (RCVYFP) must be the same. They must both specify YES or both specify NO. If SRMDEF=GLOBAL, Fast Path is always non-recoverable, so if there is a failure during the execution of a fast path transaction, the STSN session must be restarted cold.

If RCVYSTSN is specified incorrectly, message DFS1920I is issued and IMS will use the appropriate default from Table 54.

**SRMDEF=**

Specifies the status recovery mode for user and node resources. Valid values are GLOBAL, LOCAL, and NONE.

**Related Reading:** In order to understand completely what the GLOBAL, LOCAL, and NONE values mean, you have to first understand how IMS

classifies resource status. For a discussion of these concepts, see the *IMS Version 9: Administration Guide: Transaction Manager*.

The following list describes the valid values for SRMDEF.

#### **GLOBAL**

The significant status of a resource is saved globally in the coupling facility resource structure every time the significant status changes, along with all other recoverable status for that resource. The resource status is restored at the next logon or signon and is available from any IMS system in the IMSplex. Resource status is copied to the local system when that resource becomes active, but is deleted from the local system when it becomes inactive. Status is not recovered in local IMS log records.

**Requirement:** Using a status recovery mode of GLOBAL requires a Resource Manager and a resource structure in the coupling facility.

#### **LOCAL**

The significant status is saved in local control blocks and log records, along with all other recoverable resource status. The resource status is restored at the next logon or signon provided that the user or node returns to the same local IMS system. This status is not available on any other IMS system in the IMSplex.

LOCAL mode enforces an affinity for the user or node to the IMS system where the local status exists. This is referred to as RM affinity. IMS will prevent the user or node from accessing any other IMS while local status exists. If the IMS system that has local status fails, the node and user will be allowed access to another IMS system. In this case, the node or user does not recover any status and when the failed IMS restarts, the local status that existed is deleted.

Using a status recovery mode of LOCAL does not require a Resource Manager. If RM is not used to manage node or user resources, then no RM affinity is enforced.

**NONE** The significant status is not saved by RM or in local log records. When the user signs on or the node logs on, the corresponding significant status does not exist.

### **User Descriptor Examples**

The following is an example of a user descriptor:

```
U SMITH ASOT=20 LTERM=(SECLT1) AUTLGN=SEC01LU2
```

## **Specifying the Number of Hash Table Slots**

The number of hash table slots built by IMS for a statically defined resource (such as LTERMs, physical terminals, and users) is one-eighth the total number of the resource. To specify additional slots required for a resource created dynamically (conversations, LTERMs, physical terminals, and users), as a rule-of-thumb use one-eighth of the estimated total number of a dynamic resource. You might need to adjust this value. For information on setting the number of slots for specific hash tables, see the LHTS, NHTS, and UHTS parameters on the IMS and DCC procedures.





- "Second parameter may be missing"
- "First parameter exceeds three digits"
- "Delimiter is not a comma"
- "Partition name must start with an alpha"
- "Delimiter is not a close parenthesis"
- "Partition name exceeds seven characters"
- "Invalid character in partition name"
- "Card contains all spaces"
- "Invalid card input"
- "Space must follow close parenthesis"
- "First parameter missing"
- "Comma and partname missing"
- "Partition name is missing"
- "Partition name starts with numeric"
- "First parameter must not be zero"
- "Comment card"

After all of the statements are validated, the job abnormally terminates with an abend code of U0201.

---

## Chapter 9. Accessing IMS Databases through the ODBA Interface

Open Database Access (ODBA) provides a callable interface that enables any z/OS recoverable, resource-managed z/OS address space to issue DL/I database calls to an IMS DB subsystem.

The interface provides z/OS application programs (hereafter called ODBA applications) access to full-function DL/I databases and data entry databases (DEDBs). The ODBA application and IMS must coexist on the same z/OS image.

Application programs that use the ODBA interface must use the APPLCTN macro to define the PSB names required by the ODBA applications.

The ODBA interface allows IMS DB and ODBA application programs to be developed, installed, and maintained independently. This independence provides failure isolation and resource recovery by using z/OS Resource Recovery Services (RRS).

**Related Reading:** For a description of RRS/z/OS, its uses, and a discussion of Distributed Sync Point, see *IMS Version 9: Administration Guide: Transaction Manager*.

The ODBA interface resides in the z/OS address space and is recognized by IMS as an application region (hereafter called the z/OS application region).

The ODBA interface uses the Database Resource Adapter (DRA) to communicate with IMS DB.

---

### Setting Up the DRA and the ODBA Interface

To set up ODBA and the DRA, follow these steps:

1. Create the ODBA DRA startup table.
2. Place the ODBA and DRA modules in the STEPLIB or JOBLIB in the z/OS application region.
3. Bind the ODBA application programs with DFSCDLI0.
4. Set up security.

### Creating the ODBA DRA Start-up Table

Create the ODBA DRA startup table using the DFSPRP macro.

The startup table uses the DSNAME to dynamically allocate the data set that contains the rest of the ODBA interface routines. The DDNAME is generated to allow multiple connections to IMS from the same z/OS application region. If you specify the DDNAME on the DFSPRP macro, it is ignored.

The default DSNAME is IMS.SDFSRESL. This is the default name established by the IMS definition process. Make sure this data set is APF authorized.

**Note:** IMS.SDFSRESL library does not contain an DRA startup table. You must generate your own table by using the DFSPRP macro. Name the load module based on the following naming conventions:

- Characters 1-3 = DFS
- Characters 4-7 = specified 4-byte ID  
The 4-byte ID should be the IMSID of the IMS system to which you will connect. However, this is not a requirement.
- Character 8 = 0 (zero)

Ensure that the DRA startup table module name is not the same as the name of an existing IMS module. To prevent accidental overlay, put the module in a load library that is accessible by the z/OS application region and not by the IMS region.

If you use a different library for your own versions, make sure that the library is APF authorized. The DRA callable interface dynamically allocates the library by using the data set name specified in the DRA startup table. The DDNAME is generated to allow multiple connections to IMS from the same z/OS application region.

## Loading and Running the ODBA and DRA Modules in the z/OS Application Region

Place the following ODBA and DRA modules in the STEPLIB or JOBLIB in the z/OS application region. These modules are shipped with IMS in IMS.SDFSRESL.

- DFSCDLI0** This module is bound or loaded by an application program. DFSCDLI0 also contains the ALIAS name AERTDLI.
- DFSAERG0** This module is loaded by DFSCDLI0.
- DFSAERM0** This module is attached by DFSAERG0 in the z/OS application region.
- DFSAERA0** This module is attached by DFSAERM0 for initialization to the specified IMS DB subsystem.

## Binding Application Programs

Bind the ODBA application programs with DFSCDLI0 (AERTDLI). As an alternative, you can issue a load and branch command passing the AIB call list in Register 1.

**Related Reading:** For information about the AERTDLI interface, see the *IMS Version 9: Application Programming: Database Manager*.

## Establishing and Defining Security

IMS provides several options for establishing and defining security for application programs that use the ODBA interface. The options that you select depend on the type of security environment and authorization method that you plan to use. In general, the process that IMS uses to secure PSBs involves one of the following types of security checking:

- Application group name (AGN) security  
When a PSB is allocated, a security check is performed to determine:
  - If the user is authorized to connect to the AGN name assigned to the dependent region
  - If the PSB name is defined in the AGN table entry with the authorized AGN name

- Resource access security (RAS)  
A security check is performed by RACF to determine if the user is authorized to use the PSB. RACF determines authorization by looking at the RACF security class profile defined for the dependent region.

- APSB security

A security check is performed to determine if the user is authorized to use the PSB.

Table 55 identifies the values that you need to specify to control data access for specific security implementations. The table also indicates the type of security checking that is performed for each set of specifications.

Table 55. Options for Controlling Data Access for Applications that Use ODBA

Security Implementation	Authorization Method	ISIS= Specification	ODBASE= Specification	Connection Security	PSB Security
AGN security	RACF	1	N	X	X
	User Exit DFSISIS0	2	N	X	X
Resource access security	RACF	R	N		X
	Exit routine <sup>1</sup> DFSRAS00	C	N		X
	RACF and exit routine DFSRAS00 <sup>1</sup>	A	N		X
	None	0   N	N		
APSB security	RACF	Not applicable	Y		X

**Notes:**

1. Resource Access Security exit routine (DFSRAS00)

To control data access for application programs that use the ODBA interface, follow the techniques discussed in the following sections to establish connection security and PSB security checking.

**Related Reading:** For more information about IMS security options, see the *IMS Version 9: Administration Guide: System*. For more information about the Resource Access Security exit routine, see the *IMS Version 9: Customization Guide*.

**AGN Security**

Use the IMS ISIS execution parameter to control the authorization for an z/OS application region to connect to the IMS DB environment. If you specify ISIS=1 or ISIS=2, both the z/OS application region connection and PSB scheduling are checked. If you specify ISIS= 0, neither is checked. Table 56 describes the actions that you need to perform to set up security when specific options are selected.

Table 56. Options for Defining AGN Security for Applications that Use ODBA

Specification	Actions to Perform
ISIS=0	No action required. No connection security checking is performed.

Table 56. Options for Defining AGN Security for Applications that Use ODBA (continued)

Specification	Actions to Perform
ISIS=1	<ol style="list-style-type: none"> <li>1. Before the z/OS application region connects to IMS DB, make sure that the user ID and application group name (AGN) from the DRA Startup Table are authorized to access the specified IMS DB.</li> <li>2. Build RACF tables to define valid user ID-AGN combinations. If the DRA Startup Table values do not correspond to an entry in RACF's tables, the z/OS application region cannot connect to IMS DB.</li> <li>3. Use the AGN and AGPSB statements to define the AGN name and PSB name to the Security Maintenance utility.</li> </ol> <p><b>Note:</b> Connections to different IMS DB systems from the same JOB will not use the same user ID-AGN security because different DRA Startup Table load modules will be loaded.</p>
ISIS=2	<ol style="list-style-type: none"> <li>1. Create an Application Group Name exit routine that is named DFSISIS0. This routine must determine whether the AGN passed to it is valid for the attempted connection.</li> <li>2. Use the AGN and AGPSB statements to define the AGN name and PSB name to the Security Maintenance utility.</li> </ol>

### RAS Security

Use the ISIS and ODBASE execution parameters to control the authorization for an z/OS application region to use a PSB. Table 57 describes the actions that you need to perform to set up security when specific options are selected.

Table 57. Options for Defining RAS Security for Applications that Use ODBA

Specifications	Actions to Perform
ISIS=0   N and ODBASE=N	No action required. No PSB security checking is performed.
ISIS=1   2 and ODBASE=N	Perform the actions described in Table 56 on page 467 for ISIS=1 or ISIS=2.
ISIS=R and ODBASE=N	Define the PSBs that you want protected by RACF to the AIMS or Axxxxxxx resource class, and then define the user IDs that you want authorized to access the PSBs. The ODBA support for IMS will use the security environment (ACEE) passed in the dependent region's task (TCBSENV), if present, or the dependent region's address space (ASXBSENV), if the ACEE is not present at the task level.

Table 57. Options for Defining RAS Security for Applications that Use ODBA (continued)

Specifications	Actions to Perform
ISIS=C and ODBASE=N	Create a Resource Access Security exit routine that is named DFSRAS00. This routine must determine if the user is authorized to use the PSB.
ISIS=A and ODBASE=N	<ol style="list-style-type: none"> <li>1. Define the PSBs that you want protected by RACF to the AIMS or Axxxxxxx resource class, and then define the user IDs that you want authorized to access the PSBs. The ODBA support for IMS will use the security environment (ACEE) passed in the dependent region's task (TCBSENV), if present, or the dependent region's address space (ASXBSENV), if the ACEE is not present at the task level.</li> <li>2. Create a Resource Access Security exit routine that is named DFSRAS00. This routine must determine if the user is authorized to use the PSB.</li> </ol> <p>RACF is called first, and then the exit routine is called.</p>

**APSB Security**

Use the ODBASE execution parameter to control the authorization for a user to use a PSB. Table 58 describes the actions that you need to perform to set up security when specific options are selected.

Table 58. Options for Defining APSB Security for Applications that Use ODBA

Specifications	Actions to Perform
ODBASE=Y	<ol style="list-style-type: none"> <li>1. Define the PSBs that you want protected by RACF to the AIMS or Axxxxxxx general resource class (where xxxxxxx is the value specified on the RCLASS= parameter of the IMS SECURITY macro).</li> <li>2. Specify RCLASS=IMS   xxxxxxx and TYPE=RACFAGN   RACFCOM   RACFTERM   RASRACF   RAS on the IMS SECURITY macro at IMS system definition time.</li> </ol>

**ICMD Security**

For ICMD authorization, the ODBA application must set AIBRSNM1 field to the PSBNAME that was used on the previous APSB call. The PSBNAME is used by the existing ICMD support routines to issue the RACF call for security processing.





---

## Chapter 10. Accessing IMS Databases with CICS

### **In this Chapter:**

- “Installing Data Sharing with CICS”
- “Installing CICS for Use with IMS Intersystem Communication”
- “Coding Considerations for PSBs”
- “Using Sequential Buffering” on page 472
- “CICS Connected to DL/I” on page 473
- “Accessing DBCTL Databases with CICS as a CCTL” on page 473

CICS can access DL/I databases via DBCTL. This means that DBCTL satisfies CICS DL/I requests by means of the CICS-DBCTL interface.

CICS/ESA and CICS Transaction Server for OS/390 provide support for IMS Version 9 DBCTL.

---

### Installing Data Sharing with CICS

CICS users can optionally use the IMS data sharing facility. The Internal Resource Lock Manager (IRLM) is mandatory for block-level data sharing, but not for database-level sharing.

**Related Reading:** For additional information, refer to *IMS Version 9: Administration Guide: System*.

---

### Installing CICS for Use with IMS Intersystem Communication

**Related Reading:** Information on defining CICS as an Intersystem Communication node and defining CICS tables to be compatible with IMS in an Intersystem Communication network is provided in *IMS Version 9: Administration Guide: Transaction Manager*.

---

### Coding Considerations for PSBs

This section provides some guidelines for coding PSBs.

**Related Reading:** For basic information on coding program specification blocks (PSBs), refer to *IMS Version 9: Utilities Reference: System*.

**PSBs for Online Transactions:** A PSB is needed for each online program that accesses DL/I databases.

- The name of the PSB specified in the PSBNAME= keyword of the PSBGEN macro must be exactly the same as one of the entries in the PDIR.
- The name of the PSB must also be the same as the name specified in the scheduling call issued by an online transaction. If the online transaction does not specify any name in the scheduling call, the name of the PSB must be exactly the same as the name of the program associated with the transaction in the CICS program control table (PCT).

**Example:** Assume transaction X is associated with program Y. If program Y links or transfers control to program Z, which issues the scheduling call without specifying any PSB name, the default name for the PSB is Y, because it is program Y that is associated with transaction X in the PCT.

- The CMPAT=YES option on the PSBGEN statement can be omitted.
- There is no specific maximum size for an individual PSB that can be used in CICS/ESA. However, a limitation is set by the PSB pool size in DBCTL.

#### Additional Processing Intent Options (PROCOPT)

Two additional types of processing intent can be specified with the PROCOPT= keyword of the PCB or SENSEG statement. The two additional options are O and E.

PROCOPT=O specifies “read without integrity”; no dynamic enqueue is done by resource lock management for calls against the database. With the G intent option, you can specify GO, GOP, GON, or GOT. This option is only valid for the PCB statement. Read note number 1 under **Attention** for more information.

PROCOPT=E forces exclusive use of this database or segment by the online transaction. Other application programs scheduling a PSB referring to this database/segment wait during their scheduling process. No dynamic enqueue by resource lock management is done, but dynamic logging of database updates will be done. PROCOPT E can be specified with G, I, D, R, and A.

#### **Attention:**

1. The PROCOPT=O option affects integrity in reading and using uncommitted data. When you specify this option, IMS does not check the ownership of the segments returned. This means that the read-only user might have access to a segment that had been updated by another user. If the updating user then abends and is backed out, the read-only user would have seen a segment that no longer exists in the same form in the database. Consequently, if you specify this option, do not use the data that is read as a basis for updating records in any database.
2. An abend might occur with PROCOPT=GO if another program updates pointers when this program is following the pointers. Pointers are updated during insert, delete, and backout operations.
3. If PROCOPT=O in the PCB statement, the SENSEG statement must not specify a PROCOPT of I, R, D, H, or A.
4. If the O or E option is used, it must be coded immediately after the associated function code; for example GO, not OG.

---

## Using Sequential Buffering

You can use sequential buffering with CICS. To do so, you need to do two things:

- Put an SBONLINE control statement in the //DFSVSMxx file. SBONLINE allows sequential buffering to be used. For more information on SBONLINE, see “Specifying Sequential Buffering for an Online System” on page 395.
- Specify programs that are to use sequential buffering. You can do this by coding during PSBGEN an SB= keyword on the PCB macro.

**Related Reading:** For more information on the SB= keyword, see *IMS Version 9: Utilities Reference: System*.

---

## CICS Connected to DL/I

CICS can provide DL/I database support by using IMS. There are two ways in which you can use DL/I support with CICS:

- Through CICS remote DL/I support, also known as function shipping
- Through DBCTL

For CICS/ESA, remote DL/I support and DBCTL support are included in the pregenerated CICS.

---

## Accessing DBCTL Databases with CICS as a CCTL

This section gives an overview of how to connect CICS to DBCTL.

**Related Reading:** See *IMS/ESA CICS-IMS Database Control Guide* for additional details.

## DBCTL Tasks

This section describes DBCTL tasks associated with using DBCTL and CICS.

### Defining CICS resources

If you use DBCTL exclusively, define the PSBs and DMBs using the APPLCTN and DATABASE macros.

If you want to function-ship requests to a remote CICS system, in which the database manager can be DBCTL or remote DL/I (function-shipping), you need to generate a PDIR only.

**Related Reading:** Refer to *CICS System Definition Guide* and *CICS/ESA Resource Definition (Macro)* for information on defining PDIRs and DDIRs.

CICS routes DL/I requests to remote DL/I or DBCTL according to the PSB that is named. If the PSB does not appear in the CICS PDIR, and CICS is connected to DBCTL, CICS routes the request to DBCTL.

### Initializing DBCTL

Use the procedure library member DBC to initialize the DBCTL subsystem. Also generated are procedures for DBRC and DL/I, which are used to initialize the DBRC and DL/I address spaces. The DBRC and DL/I procedures are started automatically by DBCTL during DBCTL address space initialization. All three procedures use positional parameters on the EXEC statement.

```
PARM='region type, parm1, parm2, parm3,...'
```

The region types specified are:

```
PARM='DBC' for DBCTL
PARM='DRC' for DBRC
PARM='DLS' for DLISAS
```

When all three address spaces have been initialized successfully, DBCTL issues:  
DFS989I DBCTL READY

When the “READY” message is received, the IMS console operator enters a /START command. The commands options are:

- /NRESTART CHECKPOINT 0 for a coldstart with no previous shutdown

- /NRESTART for a warmstart
- /ERESTART for an emergency restart after a failure

## CICS Tasks

This section describes CICS tasks associated with using DBCTL and CICS.

### Defining DRA resources

The database resource adapter (DRA) is the interface between DBCTL and the transaction management subsystem. You need to put the DRA startup parameter table (DFSPZPxx) and the DRA startup router program (DFSPRRCO) in the CICS STEPLIB data set.

#### **Example:**

```
//STEPLIB DD DSN=CICS310.LOADLIB1,DISP=SHR
//          DD DSM=IMS.SDFSRESL,DISP=SHR
```

**Related Reading:** For more information on the DRA startup table see “Supporting CCTL Users with DBCTL Databases” on page 446, or *IMS Version 9: Customization Guide*.

### Connecting CICS to DBCTL

After CICS has been started, the CICS operator can issue a CONNECT command to DBCTL through the CDBC transaction. This transaction is also used for disconnecting from DBCTL. The CDBI transaction provides the status of the connection.

**Related Reading:** For more information on these transactions, see *IMS/ESA CICS-IMS Database Control Guide*.

---

## Chapter 11. Accessing External Subsystems with IMS

MPP, BMP, and IFP application programs can access external subsystem data, such as DB2 databases, when you use the External Subsystem Attach Facility. JMP and JBP applications can access DB2 UDB for z/OS databases when you use the DB2 Recoverable Resource Manager Services Attach Facility.

This chapter describes the system definition and tailoring tasks required to attach an external subsystem to IMS.

- “Defining Your External Subsystems to IMS” on page 475
- “Defining the Language Interface Module” on page 481
- Specifying ESSs for different regions
  - “Specifying DB2 Groups for IMS Online Regions” on page 482
  - “Specifying DB2 for IMS Batch Regions” on page 483
- “Preparing DB2 for Use with IMS” on page 483

---

### Installation Tasks for the IMS User

This section describes the installation tasks that you need to perform if you are an IMS user to connect to an external subsystem such as DB2. To access DB2 databases with IMS, follow the procedures described in this section.

#### Defining Your External Subsystems to IMS

In order for IMS to initiate contact with an external subsystem, it is necessary to add a member to the IMS procedure library, IMS.PROCLIB. This member contains an entry for each external subsystem with which IMS communicates. When defining DB2 databases, this member contains an entry for each DB2 subsystem, or an entry that defines a group of DB2s.

Using the IMS.PROCLIB data set allocated prior to Stage 2 of IMS system definition and the z/OS utility IEBUPDTE, create the PROCLIB member(s). The first 1 to 4 characters of this PROCLIB member’s name must match the IMSID, as specified by the IMSID parameter on the IMSCTRL macro statement or overridden using the execution parameter IMSID=. The last 4 positions represent any unique installation-defined identifier. This identifier is appended to the IMSID to construct the member name.

Each entry within the PROCLIB member consists of a blocked or unblocked 80-character record. Information for each record begins in position 1 of the record.

#### Specifying Member Entry Parameters

Some of the keyword parameters are required and are applicable for any external subsystem; other parameters are optional and can be unique to the external subsystem type. The SSM PROCLIB member entry parameters are listed in “SSM PROCLIB Member Entry Keyword Parameters” on page 476. Each parameter description includes information on whether the parameter is required or optional. Fields within each entry are positional and are delimited by commas. Two consecutive commas denote the absence of an optional field.

The following are guidelines for specifying member entries:

- You can specify certain parameter values with alphanumeric characters (beginning with A-Z, \$, #, or @, and containing alphabetic characters or the digits 0-9).
- Specifying duplicate subsystem names within a member is invalid. IMS uses the first definition for a specific subsystem name.
- IMS issues DFS3600I messages if it encounters any errors during SSM PROCLIB member processing. (For more information, see the *IMS Version 9: Messages and Codes, Volume 1* or *IMS Version 9: Messages and Codes, Volume 2*.)
- Use a comma to delimit keywords parameters. You can code the parameters in free format with leading and trailing blanks.
- You can have keyword format statements be continued by placing a non-blank character in position 72. The last character prior to position 72 must be a comma (to delimit the preceding parameter). Continue a statement starting in positions 2-72; a continued statement cannot start in position 1.
- You can add a comment by placing '\*' in position 1.

The SSM member needs to be defined in IMSVS.PROCLIB using a naming convention of a IMSID+SSMID.

For example, if in your IMS environment you bring up your IMS system defined with the following parameters: IMSID=SYS3 and SSM=V51A. In IMSVS.PROCLIB, there's an SSM member named SYS3V51A. This member is defined with the following values:

```
V51A,SYS1,DSNMIN10,,R,.
```

These values align with the SSN, LIT, ESMT, RTT, REO, and CRC positional parameters values.

### SSM PROCLIB Member Entry Keyword Parameters

Use the following keyword parameters to define your external subsystems to IMS. The SST= parameter must begin in position 1, signaling that keyword format is being used.

#### SST=

Is a **required** one- to eight-character, left-justified name of the *subsystem type* to which IMS connects. The SST= parameter identifies an external subsystem and can be used to identify processing that is unique to that type of external subsystem. The SST= parameter also indicates the start of a new subsystem definition and must begin in position 1. This parameter is **required** for all PROCLIB member entries defined in keyword format.

For information on the required and optional parameters for specific external subsystems, see "Parameters for Specific External Subsystems" on page 479.

#### SSN=

Is a **required** one- to four-character alphanumeric *z/OS subsystem name* of the external subsystem that will be attached. This name is specified by your installation, and must be identified to z/OS during z/OS system definition.

The external subsystem does **not** have to be an z/OS subsystem. However, if it is **not** a subsystem, some external subsystem processing functions cannot be used, such as Subsystem Interface (SSI) and End of Task/End of Memory (EOT/EOM) processing.

#### LIT=

Is a **required** (when using ESAF) one- to four-character alphanumeric field

specifying the *language interface token*. The language interface provides routing information to IMS and also contains external subsystem-dependent information for the IMS applications. IMS provides a language interface module named DFSLI000, which uses a token value of SYS1. To use this language interface, the token value SYS1 is specified for the LIT parameter to relate this language interface to an external subsystem. If you do not want to use this supplied language interface module, or want to use more than one (for example, to access test and production subsystems), assemble and link your user-defined language interface, with a token value other than SYS1, into an installation-selected library.

This token must then be specified on a record in conjunction with or instead of the standard language interface token, SYS1. Each IMS control or dependent region can thus access alternative external subsystems via the corresponding language interface module. Each individual application program, however, can be bound to only one language interface module, and can thus access only one external subsystem. Each application programmer must include the appropriate language interface module during the bind process for each application program that calls the external subsystem from an MPP, IFP, or BMP dependent region.

For information on binding the language interface module into an appropriate IMS library or defining your own language interface module, see “Defining the Language Interface Module” on page 481.

This parameter is ignored if COORD=RRS is specified.

#### **ESMT=**

Is a **required** (when using ESAF) one- to eight-character alphanumeric parameter specifying the name of the *external subsystem module table*. This table is defined in the external subsystem and specifies the external subsystem modules IMS must load, and the work areas that IMS must create.

It is the responsibility of the external subsystem to create the external subsystem module table and to make the name of that table known to the IMS user who will define these PROCLIB member entries.

This parameter is ignored if COORD=RRS is specified.

#### **COORD=RRS**

Is a required parameter when using RRSF to attach a DB2 subsystem for JMP or JBP regions. When you specify this parameter, LIT=, ESMT=, RTT=, and REO= are ignored.

#### **RTT=**

Is an **optional** one- to eight-character alphanumeric parameter specifying the name of the *resource translation table*. This table maps an application program name to the name used by the external subsystem for the entity that defines the resources needed by the application (if these names are not the same). The resource translation table need only be used if IMS and the external subsystem use different names for the same resource.

It is the responsibility of the external subsystem to create the resource translation table and to make the name of that table known to the IMS user who defines these PROCLIB member entries.

This parameter is ignored if COORD=RRS is specified.

#### **REO=**

Is an **optional** one-character *region error option* code. The option specified here determines the action IMS takes when an application program issues a request for external subsystem services before connection to the external

subsystem is complete or if problems are encountered with the external subsystem. IMS does not use these values, but passes them to the external subsystem for its use. The possible values for this command are:

- R** The application program receives a return code indicating that the request for external subsystem services has failed. If no other value is specified for the region error option, R is the default.
- Q** IMS abnormally terminates the application program with an abend code of U3051. IMS then places the transaction on a queue to be processed when the connection to the external subsystem is established. After a transaction is on this queue, it can be recovered across warm starts and emergency restarts of IMS.
- A** IMS abnormally terminates the application program with an abend code of U3047 and discards the transaction input.

This parameter is ignored if COORD=RRS is specified.

### CRC=

Is an **optional** one-character command recognition character. Any EBCDIC value except / is permitted (/ is reserved for IMS). This parameter is specified if you want to allow external subsystem commands to be entered from IMS terminals or automated operator interface (AOI) applications. Specify this parameter to allow DB2 commands to be entered from IMS terminals, DBCTL operator, or AOI applications. Commands to be executed in the external subsystem are entered at an authorized IMS terminal or by an AOI application by first entering /SSR, followed by the command recognition character specified here, followed by the external subsystem command. When IMS receives a command in this format, it routes that command to the external subsystem for processing.

**Related Reading:** For additional information on the command, see the description of /SSR in *IMS Version 9: Command Reference*.

The external subsystem can require IMS user IDs and LTERM names to be authorized to issue external subsystem commands. Refer to the external subsystem documentation for information on command authorization requirements.

When accessing DB2 databases, the terminal from which the /SSR command is issued must be authorized to issue DB2 commands. If the terminal has a signed-on user, the user ID must be authorized to DB2. If no user exists, the logical terminal name must be authorized to DB2. IMS user IDs and logical terminal names are authorized to issue DB2 commands by adding them to the DB2 catalog using the GRANT SQL statement.

**Related Reading:** Refer to *IBM DATABASE 2 Version 2 SQL Reference* for details on how to authorize IMS user IDs and logical terminal names to issue DB2 commands.

To authorize z/OS consoles in a DBCTL environment to issue DB2 commands via the /SSR command, the system console name WTOR must be added to the DB2 catalog.

### SSM PROCLIB Member Example

**SST=DB2,SSN=db2name,ESMT=esmname**

**SST=DB2,SSN=ssname,ESMT=esmname**

**SST=DB2,SSN=db2name,COORD=RRS**



## Parameters for Specific External Subsystems

The following are required and optional parameters for specific subsystem types.

**Parameters for DB2 External Subsystems:** The following parameters are required for attaching DB2 subsystems using ESAF:

SST=DB2,SSN=,LIT=,ESMT=

The following parameters are optional for attaching DB2 subsystems using ESAF:

RTT=,REO=,CRC=

The following parameters are required for attaching DB2 subsystems using RRSAP:

SST=DB2,SSN=,COORD=RRS

All other parameters are ignored when attaching DB2 subsystems using RRSAP.

## Using the SSM PROCLIB Member

After you have defined a PROCLIB member, you must do the following to use the member:

1. Insert a DD statement for IMS.PROCLIB with the ddname PROCLIB in the step execution JCL.
2. Specify the correct PROCLIB member on the EXEC statement parameter SSM= or on the /START SUBSYS SSM command.

## Specifying the SSM= EXEC Parameter

The SSM= EXEC parameter is required to enable your installation to indicate which PROCLIB member contains an entry for the external subsystem you want to attach. You can specify the SSM= EXEC parameter on the EXEC procedures of that control region, or for an MPP, IFP, or BMP dependent region.

The SSM= EXEC parameter is required for an MPP, BMP, IFP, JMP, or JBP region when the subsystem member defines a DB2 group.

If the SSM= EXEC parameter is not specified at the time the IMS subsystem is initialized, the ESAF or RRSAP component is not activated. You can activate the ESAF component and enable external subsystem support in one of two ways:

- By shutting down IMS and restarting it, specifying the SSM=EXEC parameter
- By using the /START SUBSYS SSM command to inform IMS of the subsystem member. This action activates the ESAF component and attempts a connection to the external subsystems defined in the SSM PROCLIB member.

**Related Reading:** For more information on the command, see the *IMS Version 9: Command Reference*.

**Restriction:** You can activate the RRSAP component and enable DB2 support only by shutting down IMS and restarting it with the SSM= EXEC parameter specified.

For an MPP or an IFP dependent region that is currently active but not connected, you must stop and start the region for a connection for that dependent region.

When SSM= is specified for the control region, any dependent region running under the control region can attach to the external subsystems named in the PROCLIB member pointed to by the SSM= parameter.

When SSM= is specified for an MPP, BMP, or IFP region<sup>7</sup>, it provides a filter that determines whether a given application program running in that dependent region can access external subsystem resources. Only the subsystems defined in the subsystem member can be connected to the dependent region. If a subsystem is defined in the named dependent region member, but not in the control region member, a message is sent to the MTO during dependent region initialization and a subsystem connection is not made for this dependent region entry.

For example, if a subsystem entry for a given DB2 subsystem exists in the named dependent region member, but not in the control region member, then a message will be sent to the MTO during dependent region initialization and a subsystem connection will not be made for this dependent region entry.

If SSM is specified on the control region but the installation wants a particular dependent region to be prohibited from attaching to any external subsystem, a PROCLIB member must be created having **no** entries, and its ID specified in the dependent region. Application programs in this dependent region cannot attach to any external subsystem.

When the PROCLIB member contains a null record, you will receive the following informational message:

```
DFS3600I UNABLE TO INITIATE THE EXTERNAL SUBSYSTEM TABLE, RC=36
```

**Related Reading:** For an explanation of this message, see *IMS Version 9: Messages and Codes, Volume 1* or *IMS Version 9: Messages and Codes, Volume 2*.

If an invalid member name is specified or if the record format is incorrect, no external subsystem is attached, but initialization continues. You can then enter the /START SUBSYS SSM command, specifying the correct subsystem name. If the /START SUBSYS SSM command is issued with an invalid subsystem member name, re-enter the command with a valid name.

**Restriction:** You cannot use the /START SUBSYS SSM command for external subsystems attached by RRSF. If an invalid member name is specified or if the record format is incorrect, you must restart IMS with the corrected member name or record format.

If any record in the SSM PROCLIB member is invalid, a DFS3600 message is issued and processing continues. Valid records allow an external subsystem to be attached. Invalid records in the SSM PROCLIB member are ignored but can be corrected. Those external subsystems can be attached to the IMS control region using the /START SUBSYS command.

### Supported Positional Parameters

An SSM PROCLIB member can contain both keyword format statements and positional format statements in any order. However, you cannot mix keyword and positional parameters in a single subsystem definition. The following positional parameters are accepted:

```
SSN,LIT,ESMT,RTT,REQ,CRC
```

**Recommendation:** Use keyword parameters as described in “Specifying Member Entry Parameters” on page 475.

---

7. SSM= cannot be specified on the EXEC procedures of JMP or JBP dependent regions.

## Defining the Language Interface Module

In order for an IMS application program to use the language interface module, the application program modules must be bound using the IMS reenterable DL/I language interface. The true attributes must be specified; for example, if a module is reusable only, don't specify "RENT."

**Restriction:** IMS does not support dynamic calls to the language interface.

To make dynamic calls, provide the RMODE value at bind time so that the language interface will be shown below the 16MB line.

The IMS DL/I language interface is not reenterable. Any IMS application programs that were designed to be reenterable or serially reusable can use external subsystem resources only after being bound with the IMS language interface.

**Important:** IMS and DB2 share a common alias name, DSNHLI, for the language interface module. Keep this in mind when concatenating your libraries to ensure the correct order.

- If you are running an IMS dependent region with an application program using the COBOL dynamic option, the IMS library must be concatenated first.
- If you are running an application program strictly under DB2, be sure the DB2 library is concatenated first.

For subsystems other than DB2, IMS provides a language interface module, DFSLI000, which supports the external subsystem interface. DFSLI000 passes a language interface token value of SYS1. The installation can use this module or it can define its own language interface if it wants to use a LIT value other than SYS1. When two or more external subsystems are accessed by the IMS system, the installation must define its own language interface module(s) because each subsystem has a unique LIT.

IMS provides the DFSLI macro to assist the installation in generating a language interface module. The code necessary to perform the language interface function is generated in the DFSLI macro expansion. The IMS macro library must be supplied when the macro statements are compiled to generate the module. The format of this macro is as follows, where user-supplied values must be substituted for the x's.

```
DFSLIxxx DFSLI TYPE=V2DB,LIT=xxxx
        END
```

where:

### **DFSLIxxx**

Specifies the CSECT name for the module.

**Recommendation:** Make the CSECT name match the bind name of the language interface module.

For the IMS-supplied language interface module, the value for this parameter is DFSLI000.

### **TYPE=**

Specifies the language interface type to IMS. The only value that you can enter for this parameter is V2DB.

### **LIT=**

Specifies the language interface token that relates this language interface to an external subsystem by means of their respective specifications in a PROCLIB

member entry as the LIT and SSN parameters. The IMS-supplied language interface module DFSLI000 uses a LIT value of SYS1.

**Example:** When IMS issues an external subsystem request, IMS knows the target subsystem by the LIT used in the request. For example, consider the case of an IMS dependent region accessing two external subsystems ESS1 and ESS2.

- You already have a default language interface module DFSLI000, as specified by the default LIT, SYS1. You now generate a second language interface DFSLI001 with a LIT of SYS2.
- Define two entries in the PROCLIB SSM member. The first entry points to ESS1 with LIT=SYS1. The second entry points to ESS2 with LIT=SYS2.
- Bind IMS application programs accessing ESS1 with the IMS-provided language interface module DFSLI000.
- Bind IMS application programs accessing ESS2 with the language interface module that you have generated, DFSLI001.

Although a region can communicate with two or more external subsystems, an IMS application can access only the external subsystem referred to in the bound language interface. However, you can alter the SSM member to route application requests to a different external subsystem from the subsystem defined in the language interface. Alternatively, you can use the DFSESS entry point in the language interface module.

## Specifying the Desired Trace Options

If you want to trace the connection and disconnection activities of IMS to the external subsystem, the parameter SUBS= on the OPTIONS statement must be set to ON. Specifying OUT on this parameter causes trace information to be written to the log. If this parameter is omitted, traces can be turned on and off by means of the /TRACE command.

---

## Specifying DB2 Groups for IMS Online Regions

IMS supports the use of DB2 OS/390 group. The DB2 OS/390 group name can be used in all online regions. IMS uses the subsystem name from SSM member entries to build an OS/390 group name when the named subsystem is not attached to the control region. The group name is then used to call OS/390 Name Services. If OS/390 returns a positive response to the group inquiry, IMS will select a member of the DB2 group that is connected to the IMS control region and connect that DB2 to the dependent region with the following restrictions.

### **Restrictions:**

- The selected DB2 was not explicitly defined by another statement with the SSM member
- The selected DB2 was not previously selected from the OS/390 group

If either of the preceding restrictions is encountered, IMS will ignore the statement that caused the OS/390 name services call (no messages will be issued).

The following apply to SSM definitions when using DB2 groups:

- SSM= must be specified when starting the dependent region
- SSN= must specify the one to four-character DB2 group name

- When CRC= is specified and a DB2 has been selected from a group, the CRC will be changed to that of the control region's corresponding SSM member statement

---

## Specifying DB2 for IMS Batch Regions

The SSM= parameter can be used in IMS DL/I and DBB batch regions to specify the DB2 connection parameters instead of the DDITV02 DD statement.

The subsystem member in PROCLIB is the same format as for online; however, only a single DB2 can be specified. The LIT token used is SYS1, and the following additional parameters are passed to the DB2 by IMS in batch regions:

- Connection Name. IMS uses the job name for the batch region as the connection name to provide uniqueness within the OS/390A system for the DB2.
- Plan. IMS uses the application program name as the plan name. The resource translation table (RTT) can be used to translate the program name to the desired plan name if the plan has another name.
- Program Name. The application program name can be specified in one of the following ways:
  - DSNMTV01 can be specified as the application program name for the batch region. When this method is used, control is given to DSNMTV01. When the DB2 environment is established, control is passed to the application program.
  - The application program name can be specified for the batch region. When this method is used, the SSM= parameter must be specified on the batch region parameters. Control is given to DSNMTV01 directly to establish the ESS environment. When the DB2 environment is established, control is passed to the application program specified in the batch region.

The subsystem member allows a batch region to use an DB2 without specifying the DB2 parameters and application program in the batch region JCL. However, the DDITV02 DD statement can still be used. In fact, the SSM= parameter is overridden if both the DDITV02 DD statement is used and DSNMTV01 is the batch application program name.

The DDOTV02 DD statement must be used for messages and diagnostic information regardless of how the DB2 connection parameters are specified.

---

## Preparing DB2 for Use with IMS

This section describes tasks you must perform to prepare DB2 for use with IMS.

### Providing Appropriate Tables in Your ESS

It is the responsibility of the DB2 user to provide the external subsystem module table and, optionally, the resource translation table in the DB2 subsystem. It is also the DB2 user's responsibility to make the names of these tables known to the person who is responsible for defining the IMS.PROCLIB member entries. The person defining the PROCLIB entries includes the names of these tables as values for the parameters ESMT and RTT, respectively.

### Placing DB2 Modules and Tables in the Appropriate Library

IMS loads the modules and tables specified in the external subsystem module table from the DB2 libraries when the SSM EXEC parameter is specified in an IMS region. To ensure that this process is successful, you must make the DB2 libraries, which must be APF-authorized, available to IMS. For this purpose, you can either

add the DB2 libraries to the JOBLIB/STEPLIB/LINKLIST concatenation or create a DFSESL DD statement for this purpose. (IMS does not automatically generate a DFSESL DD statement.)

If the JOBLIB/STEPLIB/LINKLIST concatenation is not authorized, you must use the DFSESL DD statement. For online IMS regions, the subsystem library or libraries must be concatenated after the library containing the IMS modules (usually IMS.SDFSRESL). When multiple subsystems are connected, additional subsystem data sets can be concatenated.

**Note:** The DFSESL DD statement is used by IMS to load specific modules or tables and is not part of a general library search.

**Example:**

```
//DFSESL DD DISP=SHR,DSN=IMS.SDFSRESL
//      DD DISP=SHR,DSN=DSNxxx.DSNLOAD
//      DD DISP=SHR,DSN=DSNyyy.DSNLOAD
```

IMS.SDFSRESL is first, followed by the subsystem libraries.

For IMS batch regions, the DFSESL DD statement is not used. The DB2 libraries must be in both the JOBLIB/STEPLIB/LINKLIST concatenation and the DFSRESLB DD statement for authorized modules.

If you use the COBOL II dynamic option, the DB2 libraries must be added to the JOBLIB or STEPLIB regardless of authorization. The IMS.SDFSRESL DD statement must precede the reference to DSNxxx.DSNLOAD in the JOBLIB or STEPLIB. If the JOBLIB or STEPLIB is not authorized, you must also use a DFSESL DD statement.

## Appendix A. IMS System Definition (Step 3) Examples

The IMS system definitions in this appendix are representative samples of the stage 1 input to the assembler during an IMS system definition. The following sample definitions are presented:

- “IMS DB/DC Environment”
- “Data Communication Macro Statements” on page 490
- “IMS Multiple Systems Coupling” on page 511
- “Data-Sharing System Configuration” on page 516
- “IMS DBCTL Environment” on page 521

An example of IMS system definition stage 1 input, which defines all supported terminal types, is provided as a COPY member of IMS.ADFSMAC (VSINPUT). To obtain an up-to-date example of input to, and output from, stage 1 of IMS system definition, use the following JCL on the IMSGEN macro statement:

```
// JOB
// EXEC  ASMFC,REGION=2048K
//ASM.SYSLIB DD DSN=IMS.ADFSMAC,DISP=SHR
//ASM.SYSIN  DD DSN=IMS.ADFSMAC(VSINPUT),
//          DISP=SHR
```

The IMS system definition macro statements shown in the following example are the only statements necessary for the system definition for an OS/390 database system.

Characteristics to Define	Macro Statements Prepared	Column 72
IMS Configuration	IMSCTRL SYSTEM=(VS/2,(BATCH,DB/DC),390) IMSGEN ASM=HLASM,ASMPRT=ON, LKPRT=(XREF,LIST), JOBCTL=(4,C,T), PRTY=6,PROCLIB=YES	X X X

### IMS DB/DC Environment

A hypothetical IMS DB/DC environment is the basis for this example of an IMS DB/DC system definition. An overview of the IMS system is followed by system details and macros.

In this example, the system has the following characteristics:

- 46 application programs
- 67 transaction codes against those application programs
- 16 databases

### Data Communication Characteristics

The data communication characteristics of the hypothetical environment include:

- Four local SYSOUT line groups
- One 2740 nonswitched line group with two lines
  - The first line has two 2740 Model 1 terminals
  - The second line has two 2740 Model 2 terminals
- One 2740 non-station control line group containing one line and one terminal

- One 2741 nonswitched line group containing one line and one terminal
- One 3270 local line group containing one line and six terminals
- One 3270 local line group containing one line and two 3286 printer terminals
- One 3270 remote line group containing one line, one 3275 terminal, and one 3271 control unit supporting five terminals
- One System/3 line group containing one line, with one station supporting four terminals
- One System/7 start/stop line group containing one line, with one station supporting four terminals
- One System/7 BSC line group containing one line, with one station supporting one terminal
- Two 2741 switched line groups
- One 3275 switched line group with two lines
- Six 3270 local VTAM terminals
- Six 3270 remote VTAM terminals
- Five 3600 (or FINANCE) VTAM terminals
- Three SLU 1 VTAM terminals
- Two NTO terminals
- Two SLU 2 VTAM terminals
- Two SLU P VTAM terminals
- Two LU 6.1 VTAM terminals

## System Configuration Macro Statements

The system configuration macros in Figure 35 select IMS system functions for z/OS, allowing 5 regions to operate simultaneously. They specify 16 transaction code classes, and the number of concurrently operating subtasks optimized to equal the number of specified communication lines divided by two.

Characteristics to Define	Macro Statements Prepared	Column 72
IMS Configuration	IMSCTRL SYSTEM=(VS/2,(ALL,DB/DC),390),	X
	MAXREGN=(5,52K,A,3),	X
	MCS=13,DESC=2,MAXCLAS=16,	X
	IMSID=IMSA	
	IMSCTF SVCNO=(,254,255),	X
	APNDG=(,ZZ),	X
System Buffers and Data Sets	RDS=(LGDK,2048),	X
	CPLOG=500000	
	BUFPOOLS PSB=8000,PSBW=6000,	X
	DMB=11000,FRE=28,	X
Security Facilities	MSGQUEUE DSETS=LGDK,	X
	BUFFERS=(20,1152),	X
	SHUTDOWN=150	
	SECURITY TRANCMD=NO,TERMNL=NO,	X
	PASSWD=NO,	X
	TYPE=(NORACTRM,NOSIGNEX,NOTRANEX,NOAGN),	X
	SECLVL=(NOTRAN,NOSIGN)	

Figure 35. System Configuration Macro Statements



## Database and Application Macro Statements

The macro statements shown in Figure 36 define DL/I databases.

Characteristics to Define	Macro Statements Prepared	Column 72
DL/I Databases		
DH41SK01, DH41SK02, DH41TS01	DATABASE DBD=(DH41SK01,DH41SK02,DH41TS01)	
DI31LM01	DATABASE DBD=DI31LM01	
DI21PART, DH41SK03, DX41SK01	DATABASE RESIDENT, DBD=(DI21PART,DH41SK03,DX41SK01)	X
DI41SK01, DI41SK02	DATABASE DBD=(DI41SK01,DI41SK02)	
DD41SK01, DD41TS01, DX41TS01	DATABASE DBD=(DD41SK01,DD41TS01,DX41TS01)	
LTERMINL, LTINDXDB	DATABASE DBD=(LTERMINL,LTINDXDB)	
TESTABLE, TTINDXDB	DATABASE DBD=(TESTABLE,TTINDXDB)	

Figure 36. Database and Application Macro Statements

## Application Programs

In the following example, each DL/I application program is shown with associated transaction codes and macros.

Characteristics to Define	Macro Statements Prepared	Column 72
Programs and Transaction Codes		
HSBASK41 (Batch)	APPLCTN PSB=HSBASK41,PGMTYPE=BATCH	
HIMAJC01 TPPL1	APPLCTN PSB=HIMAJC01 TRANSACT CODE=TPPL1,PRTY=(8,8)	
HIMAJC03 TUBE	APPLCTN PSB=HIMAJC03 TRANSACT CODE=TUBE,PRTY=(8,8), SPA=100,MODE=SNGL	X
TUBFD	TRANSACT CODE=TUBFD,PRTY=(8,8), SPA=100, MODE=SNGL	X
TUBFC	TRANSACT CODE=TUBFC,PRTY=(8,8), SPA=100, MODE=SNGL	X
TUBMA	TRANSACT CODE=TUBMA,PRTY=(8,8), SPA=20000,MODE=SN GL	X
HIMALM01	APPLCTN PSB=HIMALM01, SCHDTYP=PARALLEL	X

DLI	TRANSACT CODE=DLI,PRTY=(5,10,5), PROCLIM=(10,10), MSGTYPE=(SINGLSEG,RESPONSE)	X X
ICS	TRANSACT CODE=ICS,PRTY=(5,12,5), PROCLIM=(10,100), MODE=SNGL	X X
DLN	TRANSACT CODE=DLN,PRTY=(0,8,3), PROCLIM=(10,100)	X
IMS	TRANSACT CODE=IMS,PRTY=(5,12,5), PROCLIM=(1,100), MSGTYPE=SNGLSEG,PARLIM=1	X X
CONALTR	APPLCTN PSB=CONALTR	
CONAL	TRANSACT CODE=CONAL,MODE=SNGL, PRTY=(8,8),SPA=100	X
CONALF	TRANSACT CODE=CONALF,MODE=SNGL, PRTY=(8,8), SPA=100	X X
LKMDFS00	APPLCTN PSB=LKMDFS00	
STL	TRANSACT CODE=STL,PRTY=(5,12,5), PROCLIM=(8,100),INQ=YES	X
LKMDFS10	APPLCTN PSB=LKMDFS10,PGMTYPE=BATCH	
LKM	TRANSACT CODE=LKM,PRTY=(0,0), MSGTYPE=SNGLSEG	X
MR1	APPLCTN PSB=MR1	
MR1	TRANSACT CODE=MR1,PRTY=(8,8)	
MR2	APPLCTN PSB=MR2	
MR2	TRANSACT CODE=MR2,PRTY=(8,8)	
MR3	APPLCTN PSB=MR3	
MR3	TRANSACT CODE=MR3,PRTY=(8,8)	
MR4	APPLCTN PSB=MR4	
MR4	TRANSACT CODE=MR4,PRTY=(8,8)	
MR5	APPLCTN PSB=MR5	
MR5	TRANSACT CODE=MR5,PRTY=(8,8)	
<b>Characteristics to Define</b>	<b>Macro Statements Prepared</b>	<b>Column 72</b>
Programs and Transaction Codes		
MR6	APPLCTN PSB=MR6	
MR6	TRANSACT CODE=MR6,PRTY=(8,8)	
TESTIMSD	APPLCTN PSB=TESTIMSD	
TSTCONV	TRANSACT CODE=TSTCONV,PRTY=(8,8), SPA=100,MODE=SN GL	X
TSTNORM	TRANSACT CODE=TSTNORM,PRTY=(8,8)	
REQIMSD	APPLCTN PSB=REQIMSD	
TREQ	TRANSACT CODE=TREQ,PRTY=(8,8), SPA=100,MODE=SNGL	X
LOADLT (Batch)	APPLCTN PSB=LOADLT,PGMTYPE=BATCH	
LOADTT (Batch)	APPLCTN PSB=LOADTT,PGMTYPE=BATCH	
STLECHO	APPLCTN PSB=STLECHO	

ECHO	TRANSACT CODE=ECHO,PRTY=(5,10,5), PROCLIM=(10,10), MSGTYPE=SNGLSEG	X X
ECHO	TRANSACT CODE=ECHO,PRTY=(5,10,5), PROCLIM=(10,10)	X
MR8	APPLCTN PSB=MR8	
MR8	TRANSACT CODE=MR8,PRTY=(8,8)	

<b>Characteristics to Define</b>	<b>Macro Statements Prepared</b>	<b>Column 72</b>
--------------------------------------	----------------------------------	----------------------

Programs and  
Transaction  
Codes

**RESIDENT PSBS**

RESPSB1	APPLCTN PSB=RESPSB1,RESIDENT	
RESPSB	TRANSACT CODE=RESPSB,PRTY=(8,8)	
RESBAL01	APPLCTN PSB=RESBAL01, SCHDTYP=PARALLEL,RESIDENT	X
BAL01	TRANSACT CODE=(BAL01,BAL02,BAL03), PRTY=(8,8)	X
BAL02		
BAL03		
RESBAL02	APPLCTN PSB=RESBAL02, SCHDTYP=PARALLEL	X
BALTRAN	TRANSACT CODE=BALTRAN,PRTY=(8,8), PARLIM=1	X
PSBEXINT	APPLCTN PSB=PSBEXINT	
TRANEXNT	TRANSACT CODE=TRANEXNT,PRTY=(8,8)	
INTCON	APPLCTN PSB=INTCON	
ANY	TRANSACT CODE=ANY,PRTY=(8,8),SCHD=3	
HI	TRANSACT CODE=HI,PRTY=(8,8),SCHD=2	
EQHI	TRANSACT CODE=EQHI,PRTY=(8,8), SCHD=1	X
NXCLASS	TRANSACT CODE=NXCLASS,PRTY=(8,8), SCHD=4	X
CUTOFF	APPLCTN PSB=CUTOFF	
SIX	TRANSACT CODE=SIX,PRTY=(6,6)	
SEVEN	TRANSACT CODE=SEVEN,PRTY=(7,7)	
EIGHT	TRANSACT CODE=EIGHT,PRTY=(8,8)	
NINE	TRANSACT CODE=NINE,PRTY=(9,9)	
NXTCLASS	APPLCTN PSB=NXTCLASS,PGMTYPE=(,2)	
ONE	TRANSACT CODE=ONE,PRTY=(1,1)	

The system includes one 3600 application program.

<b>Characteristics to Define</b>	<b>Macro Statements Prepared</b>	<b>Column 72</b>
--------------------------------------	----------------------------------	----------------------

Programs and  
Transaction  
Codes

TEST3600	APPLCTN DOPT,PSB=TEST3600, PGMTYPE=TP	X
CIFINQ	TRANSACT CODE=CIFINQ,PRTY=(1,2,2), MSGTYPE=SNGLSEG,	X X

	PROCLIM=(,30),INQUIRY=YES,	X
	SEGN0=256,SEGSIZE=4096	
SAVDBT	TRANSACT CODE=(SAVDBT,SAVDPT),	X
	PRTY=(1,2,2),	X
	MSGTYPE=SNGLSEG,	X
	PROCLIM=(,30),INQUIRY=NO,	X
	SEGN0=10,SEGSIZE=80	
BRNCHTOT	TRANSACT CODE=BRNCHTOT,	X
	PRTY=(1,2,2),MSGTYPE=SNGLSEG,	X
	PROCLIM=(,30),INQUIRY=YES,	X
	SEGN0=256,SEGSIZE=132	
PGM3741	APPLCTN PSB=PGM3741,PGMTYPE=TP	
TX3741	TRANSACT CODE=TX3741	
DFSSAM02	APPLCTN PSB=DFSSAM02,	X
	PGMTYPE=(TP,,4)	
PART	TRANSACT CODE=PART,PRTY=(7,10,2),	X
	INQ=(YES,NORECOV)	
DFSSAM03	APPLCTN PSB=DFSSAM03	
DSPINV	TRANSACT CODE=DSPINV,PRTY=(7,10,2),	X
	INQ=YES,MSGTYPE=(,4)	
DFSSAM04	APPLCTN PSB=DFSSAM04,PGMTYPE=(,4)	
ADDPART	TRANSACT CODE=(ADDPART,DLETPART),	X
	PRTY=(7,10,2)	
DLETPART		
ADDINV	TRANSACT CODE=(ADDINV,DLETINV),	X
	PRTY=(7,10,2),MSGTYPE=(,5)	
DLETINV		
DFSSAM05	APPLCTN PSB=DFSSAM05,PGMTYPE=(,3)	
CLOSE	TRANSACT CODE=CLOSE,PRTY=(7,10,2)	
DFSSAM06	APPLCTN PSB=DFSSAM06	
DISBURSE	TRANSACT CODE=DISBURSE,	X
	PRTY=(7,10,2),MSGTYPE=(,2)	
DFSSAM07	APPLCTN PSB=DFSSAM07,PGMTYPE=(,2)	
DSPALLI	TRANSACT CODE=DSPALLI,	X
	PRTY=(7,10,2),INQ=YES	

---

## Data Communication Macro Statements

| This section describes several line group configurations and includes the required  
| macro statements for each configuration. The macro sets that are described in this  
| section include:

- | • “BTAM, BSAM, GAM, and ARAM Macro Sets”
- | • “Switched Communication Device Macro Sets” on page 499
- | • “VTAM Macro Sets” on page 501

### BTAM, BSAM, GAM, and ARAM Macro Sets

The following macros describe the BTAM facilities. A graphic representation is presented, followed by examples of the required macros statements.

#### Local SYSOUT Line Group

Figure 37 on page 491 shows a local SYSOUT line group.

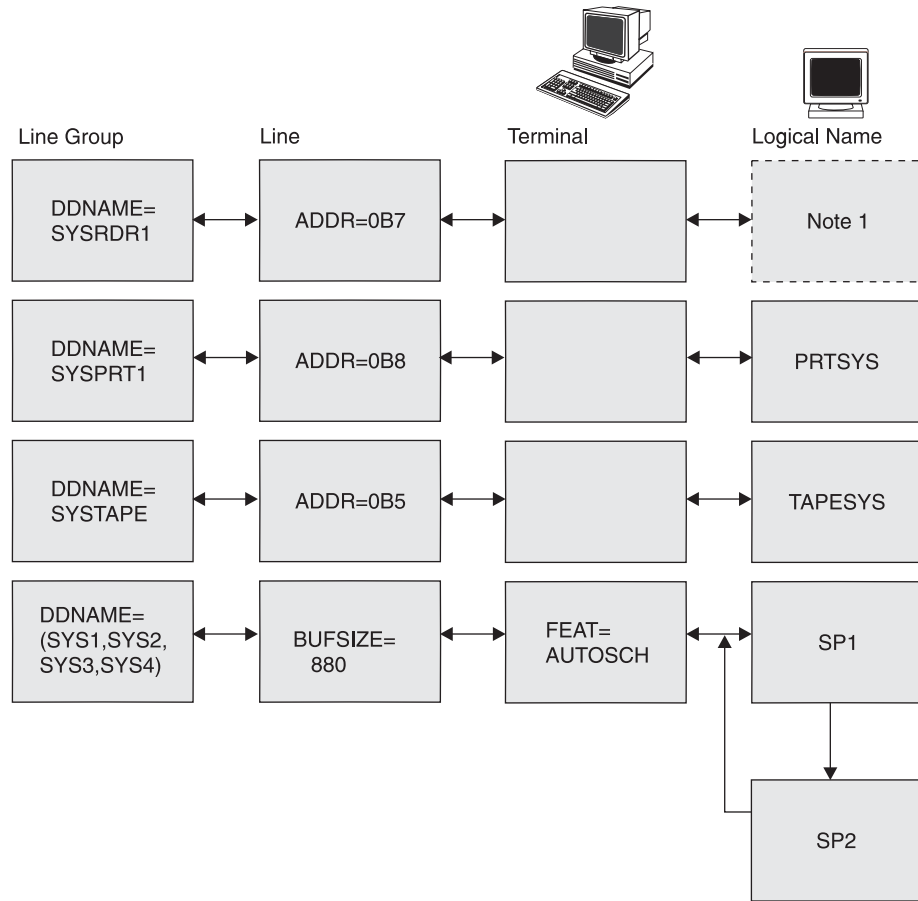


Figure 37. Local SYSOUT Line Group

**Macro Statement Set Prepare**

```

LINEGRP DDNAME=SYSRDR1,UNITYPE=READER
LINE ADDR=0B7
TERMINAL LTERM=T2741
LINEGRP DDNAME=SYSPRT1,UNITYPE=PRINTER
LINE ADDR=0B8
TERMINAL
NAME PRSYS
LINEGRP DDNAME=SYSTAPE,UNITYPE=TAPE
LINE ADDR=0B5,BUFSIZE=220
TERMINAL
NAME TAPESYS
LINEGRP UNITYPE=SPOOL,DDNAME=(SYS1,SYS2,SYS3,SYS4)
LINE BUFSIZE=880
TERMINAL FEAT=AUTOSCH
NAME SP1,SP2
    
```

Figure 38. Macro Statements for Local SYSOUT Line Group

**2740 Nonswitched Line Group**

Figure 39 on page 492 shows a 2740 nonswitched line group.

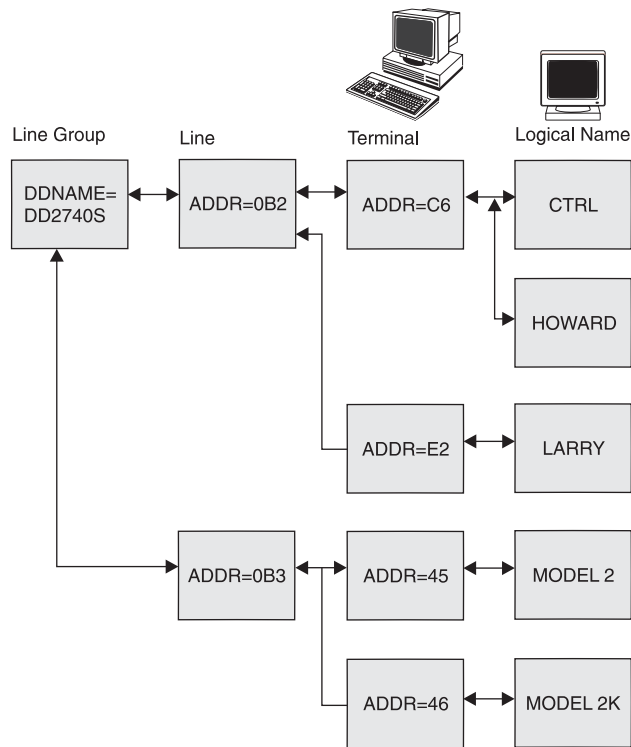


Figure 39. 2740 Nonswitched Line Group

Figure 40 shows the macro statements for the 2740 nonswitched line group.

#### Macro Statement Set Prepared

```
LINEGRP DDNAME=DD2740S,UNITYPE=2740
LINE ADDR=0B2
  TERMINAL ADDR=C6,OPTIONS=MFS
  NAME CTRL,HOWARD
  TERMINAL ADDR=E2,OPTIONS=MFS
  NAME LARRY
LINE ADDR=0B3,MODEL=(2,440)
  TERMINAL ADDR=45,BUFSIZE=248,OPTIONS=(MFS)
  NAME MODEL2
  TERMINAL ADDR=46,OPTIONS=MFS
  NAME MODEL2K
```

Figure 40. Macro Statements for 2740 Nonswitched Line Group

### 2740 Non-Station Control Line Group

Figure 41 on page 493 illustrates a 2740 non-station control line group.

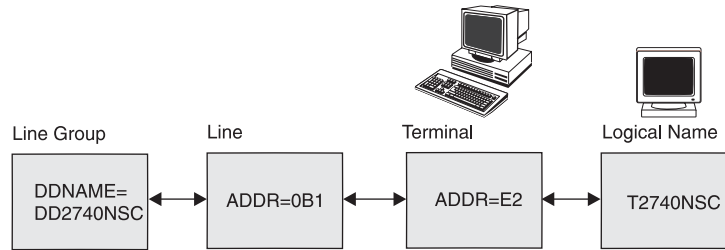


Figure 41. 2740 Non-Station Control Line Group

Figure 42 shows the macro statements for a 2740 non-station control line group.

**Macro Statement Set Prepared**

```
LINEGRP DDNAME=D2740NSC,UNITYPE=(2740,NOSTACTL)
LINE ADDR=0B1
  TERMINAL ADDR=E2,OPTIONS=MFS
  NAME T2740NSC
```

Figure 42. Macro Statements for 2740 Non-station Control Line Group

**2741 Nonswitched Line Group**

Figure 43 illustrates a 2741 nonswitched line group.

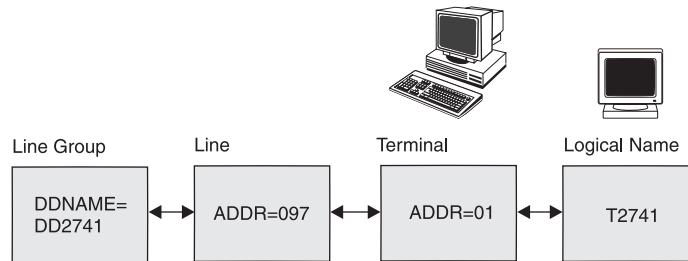


Figure 43. 2741 Nonswitched Line Group

Figure 44 shows the macro statements for a 2741 nonswitched line group.

**Macro Statement Set Prepared**

```
LINEGRP DDNAME=DD2741,UNITYPE=2741,CODE=CORRS
LINE ADDR=097
  TERMINAL ADDR=01,FEAT=NOINTRPT,OPTIONS=MFS
  NAME T2741
```

Figure 44. Macro Statements for 2741 Nonswitched Line Group

**3270 Local Line Groups**

Figure 45 on page 494 illustrates a 3270 local line group.

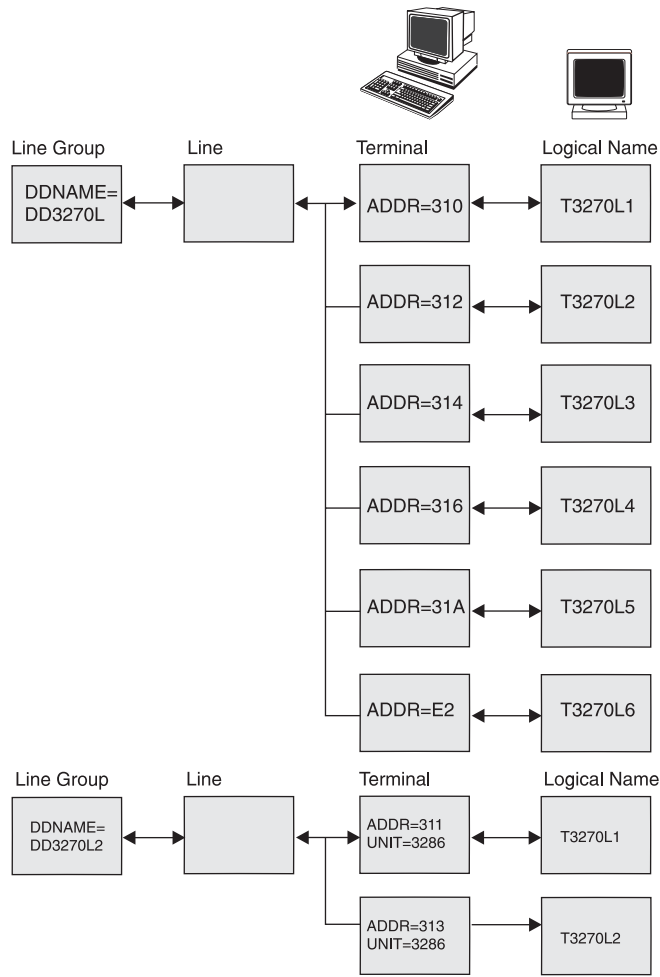


Figure 45. 3270 Local Line Groups

Figure 46 on page 495 shows the macro statements for a 3270 local line group.



Macro Statement Set Prepared	Column 72
LINEGRP DDNAME=DD3270L,UNITYPE=(3270,LOCAL)	
LINE BUFSIZE=300	
TERMINAL ADDR=310,MODEL=2	
NAME T3270L1	
TERMINAL ADDR=312,FEAT=(PFK,CARD,PEN),	X
OPTIONS=FORCRESP,MODEL=1	
NAME T3270L2	
TERMINAL ADDR=314,TYPE=3270-A1,SIZE=(12,80)	
NAME T3270L3	
TERMINAL ADDR=316,TYPE=3270-A2,SIZE=(24,80)	
NAME T3270L4	
TERMINAL ADDR=318,TYPE=3270-A5,SIZE=(12,40)	
NAME T3270L5	
TERMINAL ADDR=31A,TYPE=3270-A6,SIZE=(6,40)	
NAME T3270L6	
LINEGRP DDNAME=DD3270L2,UNITYPE=(3270,LOCAL)	
LINE	
TERMINAL ADDR=311,MODEL=2,UNIT=3286	
NAME T3270P3	
TERMINAL ADDR=313,MODEL=1,UNIT=3286	
NAME T3270P4	

*Figure 46. Macro Statements for 3270 Local Line Group*

### **3270 Remote Line Group**

Figure 47 on page 496 illustrates a 3270 remote line group.

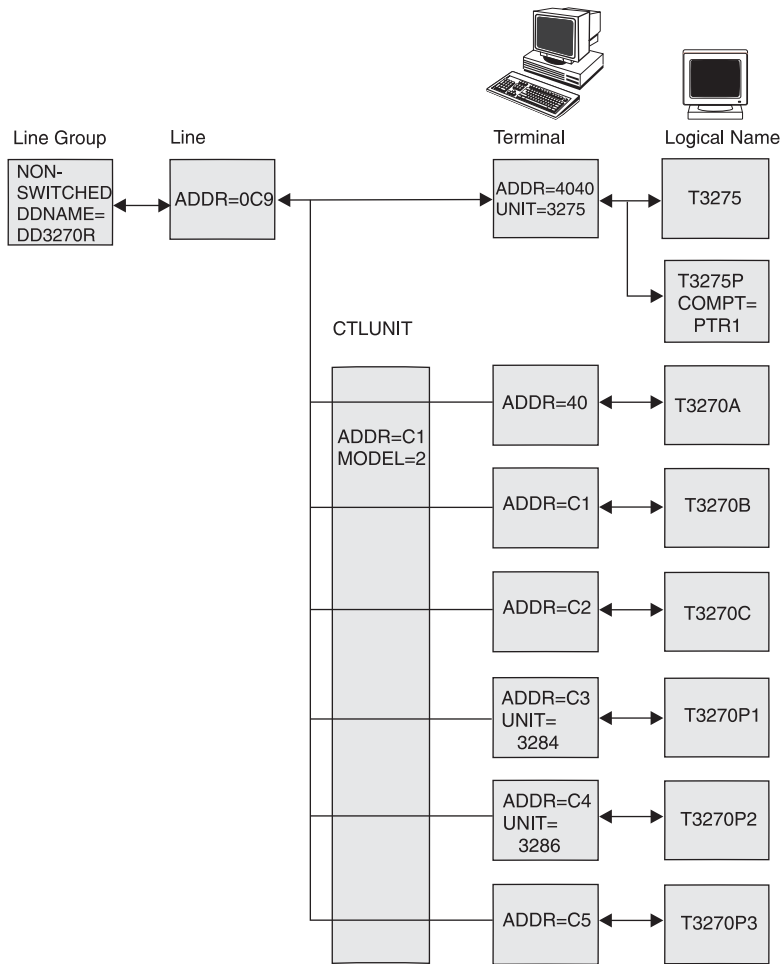


Figure 47. 3270 Remote Line Group

Figure 48 on page 497 shows the macro statements for a 3270 remote line group.

Macro Statement Set Prepared	Column
LINEGRP DDNAME=DD3270R,UNITYPE=3270,CODE=EBCDIC	72
LINE ADDR=0C9	
TERMINAL ADDR=4040,UNIT=3275,COMPT=PTR1, FEAT=PFK,OPTIONS=COPY	X
NAME T3275	
NAME T3275P,COMPT=PTR1	
CTLUNIT ADDR=C1,MODEL=2	
TERMINAL ADDR=40,FEAT=(PFK,CARD,PEN), MODEL=2,OPTIONS=COPY	X
NAME T3270A	
TERMINAL ADDR=C1,FEAT=(PFK,CARD,PEN), MODEL=2,OPTIONS=COPY	X
NAME T3270B	
TERMINAL ADDR=C2,FEAT=(PFK,CARD,PEN), MODEL=1,OPTIONS=COPY	X
NAME T3270C	
TERMINAL ADDR=C3,UNIT=3284,MODEL=2,PTRSIZE=132	
NAME T3270P1	
TERMINAL ADDR=C4,UNIT=3286,MODEL=2,PTRSIZE=132	
NAME T3270P2	
TERMINAL ADDR=C5,FEAT=F1,TYPE=3270-A2, SIZE=(24,80)	X
NAME T3270D	

Figure 48. Macro Statements for 3270 Remote Line Group

### System/3 Line Group

Figure 49 illustrates a System/3 line group.

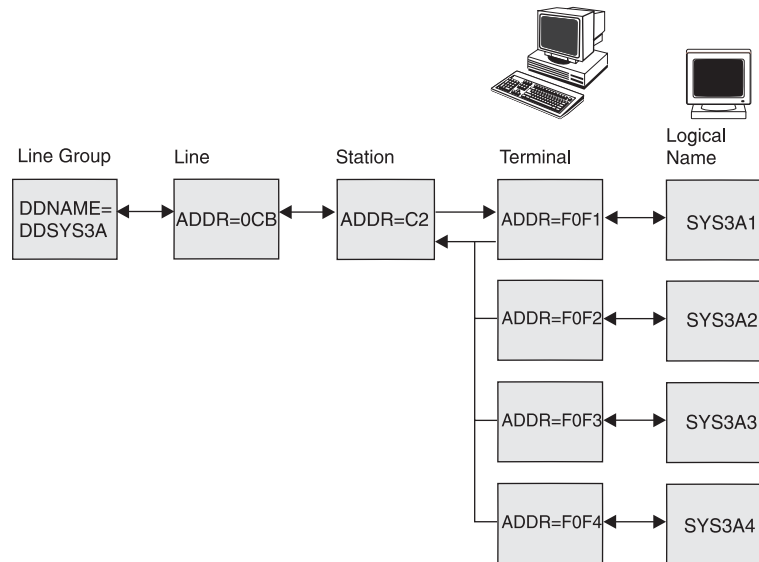


Figure 49. System/3 Line Group

Figure 50 on page 498 shows the macro statements for a System/3 line group.

**Macro Statement Set Prepared**

```

LINEGRP DDNAME=DDSYS3A,UNITYPE=S/3
LINE  ADDR=0CB,BUFSIZE=(256,256)
STATION ADDR=C2,TRANSLM=12,TYPE=ASK
TERMINAL ADDR=F0F1
NAME SYS3A1
TERMINAL ADDR=F0F2
NAME SYS3A2
TERMINAL ADDR=F0F3
NAME SYS3A3
TERMINAL ADDR=F0F4
NAME SYS3A4

```

Figure 50. Macro Statements for System/3 Line Group

**System/7 Line Group**

Figure 51 and Figure 53 on page 499 illustrate System/7 line groups.

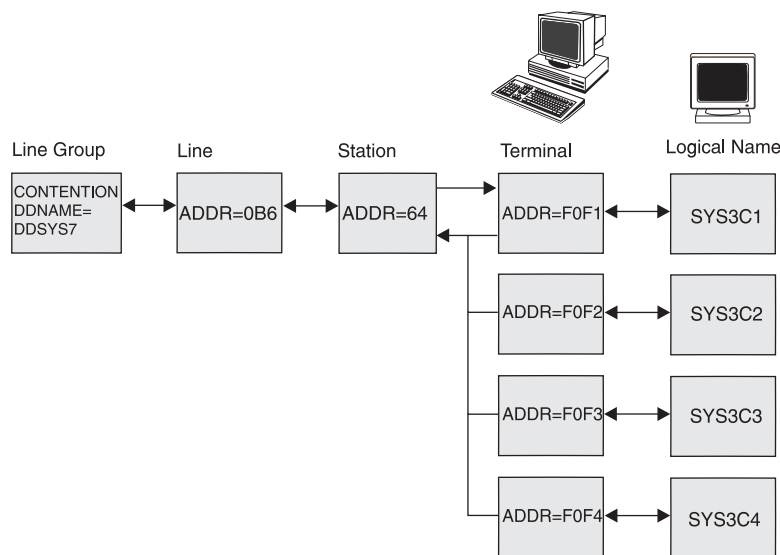


Figure 51. First System/7 Line Group

Figure 52 shows the macro statements for a System/7 line group.

**Macro Statement Set Prepared**

```

LINEGRP DDNAME=DDSYS7,UNITYPE=S/7
LINE  ADDR=0B6,BUFSIZE=(202,82)
STATION ALIGN=2,TRANSLM=12,ADDR=64,TYPE=ASK
TERMINAL ADDR=F0F1
NAME SYS7C1
TERMINAL ADDR=F0F2
NAME SYS7C2
TERMINAL ADDR=F0F3
NAME SYS7C3
TERMINAL ADDR=F0F4
NAME SYS7C4

```

Figure 52. First Macro Statements for System/7 Line Group

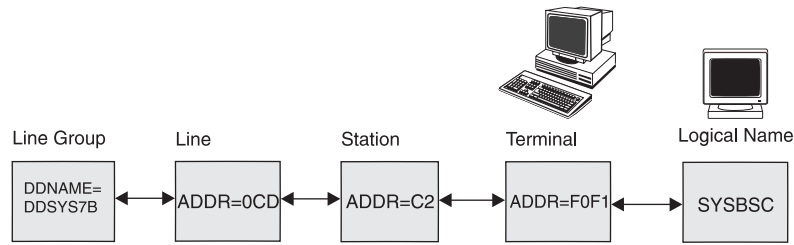


Figure 53. Second System/7 Line Group

Figure 54 shows the macro statements for the second System/7 line group.

**Macro Statement Set Prepared**

```
LINEGRP DDNAME=DDSYS7B,UNITYPE=(S/7,,BSC)
LINE ADDR=0CD,BUFSIZE=(256,256)
STATION ADDR=C2,TRANSLM=12,TYPE=ASK
TERMINAL ADDR=F0F1
NAME SYS7BSC
```

Figure 54. Macro Statements for second System/7 Line Group

## Switched Communication Device Macro Sets

The following sets of macro statements are required to describe the switched communication devices in the hypothetical system. A graphic representation of each of these is presented, along with the macro statements required to define it.

### 2740 Switched Line Group

Figure 55 illustrates a 2740 switched line group.

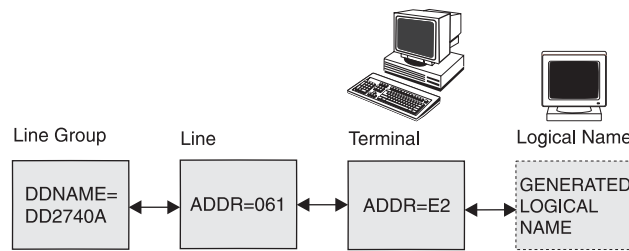


Figure 55. 2740 Switched Line Group

Figure 56 shows the macro statements to define a 2740 switched line group.

**Macro Statement Set Prepared**

```
LINEGRP DDNAME=DD2740A,UNITYPE=2740,FEAT=SWITCHED
LINE ADDR=061
TERMINAL ADDR=E2,OPTIONS=MFS
```

Figure 56. Macro Statements to Define 2740 Switched Line Group

### 2741 Switched Line Group

Figure 57 on page 500 illustrates a 2741 switched line group.

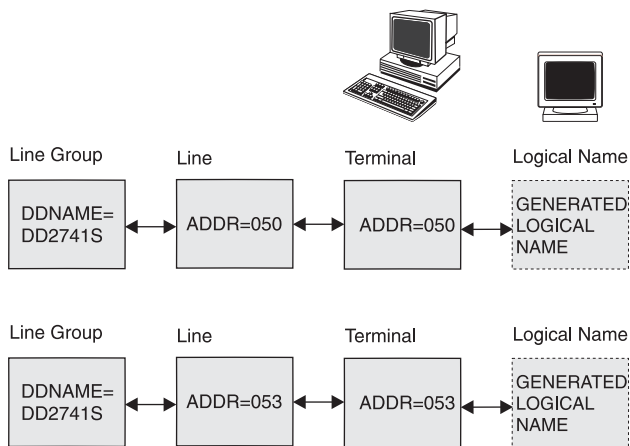


Figure 57. 2741 Switched Line Group

Figure 58 shows the macro statements to define a 2741 switched line group.

Macro Statement Set Prepared	Column
	72
LINEGRP DDNAME=DD2741S,UNITYPE=2741,	X
CODE=(CORRS,EBCDIC),FEAT=SWITCHED	
LINE ADDR=050	
TERMINAL ADDR=50,OPTIONS=MFS	
LINEGRP DDNAME=DD2741S2,UNITYPE=2741,	X
CODE=(CORRS,BCD),FEAT=SWITCHED	
LINE ADDR=053	
TERMINAL ADDR=53,OPTIONS=MFS	

Figure 58. Macro Statements to Define 2741 Switched Line Group

### 3275 Switched Line Group

Figure 59 illustrates a 3275 switched line group.

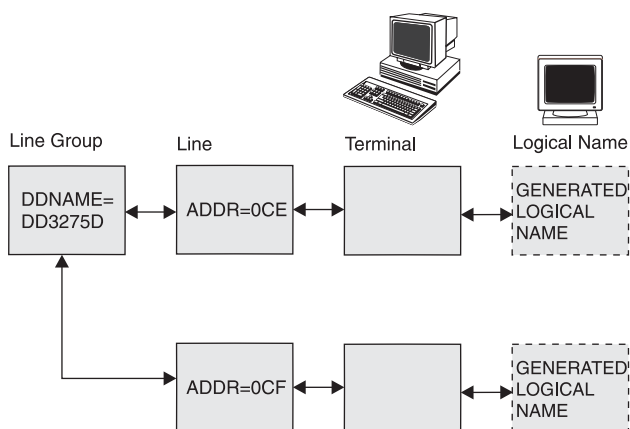


Figure 59. 3275 Switched Line Group

Figure 60 on page 501 shows the macro statements to define a 3275 switched line group.

**Macro Statement Set Prepared**

```

LINEGRP UNITYPE=3275,DDNAME=DD3275D,CODE=EBCDIC

A  CONFIG MODEL=2,FEAT=PFK,OPTIONS=NOCOPY
B  CONFIG MODEL=2,OPTIONS=NPGDEL
   LINE  ADDR=0CE,VERIFY=YES,CONFIG=B
   IDLIST (E54C5A,A),E54C5B,(E54C5C,B),(E54C5D,A)

TERMINAL

LINE  ADDR=0CF,VERIFY=NO
IDLIST (E54C5B,A),E54C5C,(E54C5D,B)

TERMINAL
    
```

*Figure 60. Macro Statements to Define 3275 Switched Line Group*

**VTAM Macro Sets**

The following sets of VTAM macro statements describe the VTAM communication devices. A graphic representation of each of these is presented, along with the macro statements required to define it.

**COMM Macro Statement**

The COMM macro statement shown in Figure 61 can only appear once in an IMS system definition. When VTAM is included in a system, the COMM macro statement is required.

Macro Statement Set Prepared	Column
COMM  RECANY=(4,500),APPLID=IMSD,	72
OPTIONS=(TIMESTAMP,MFSTEST,FMTMAST,PAGING)	X

*Figure 61. COMM Macro Statement*

**Local 3270 VTAM Terminals**

Figure 62 on page 502 illustrates macro statements for a local 3270 VTAM terminal.

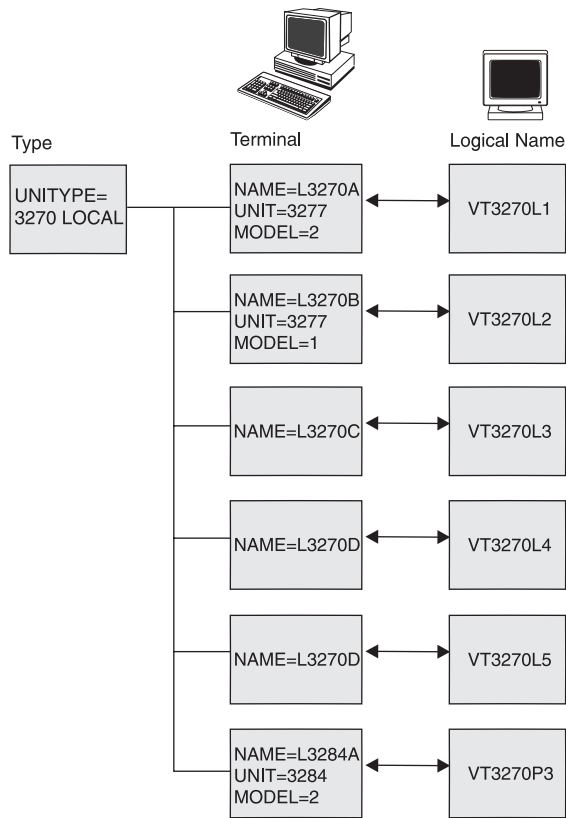


Figure 62. Local 3270 VTAM Terminal

Figure 63 shows the macro statements to define local 3270 VTAM terminals.

#### Macro Statement Set Prepared

```

TYPE UNITYTYPE=(3270,LOCAL),MODEL=2
TERMINAL NAME=L3270A
NAME (VT3270L1,MASTER)
TERMINAL NAME=L3270B,MODEL=1,OPTIONS=FORCRESP
NAME VT3270L2
TERMINAL NAME=L3270C,TYPE=3270-A2,SIZE=(24,80)
NAME VT3270L3
TERMINAL NAME=L3270D,TYPE=3270-A3,SIZE=(32,80)
NAME VT3270L4
TERMINAL NAME=L3270E,TYPE=3270-A4,SIZE=(43,80)
NAME VT3270L5
TERMINAL NAME=L3284A,UNIT=3284,PTRSIZE=132
NAME (VT3270P3,SECONDARY)

```

Figure 63. Macro Statements to Define 3270 VTAM Terminals

#### Remote 3270 VTAM Terminals

Figure 64 on page 503 illustrates macro statements for remote 3270 VTAM terminals.



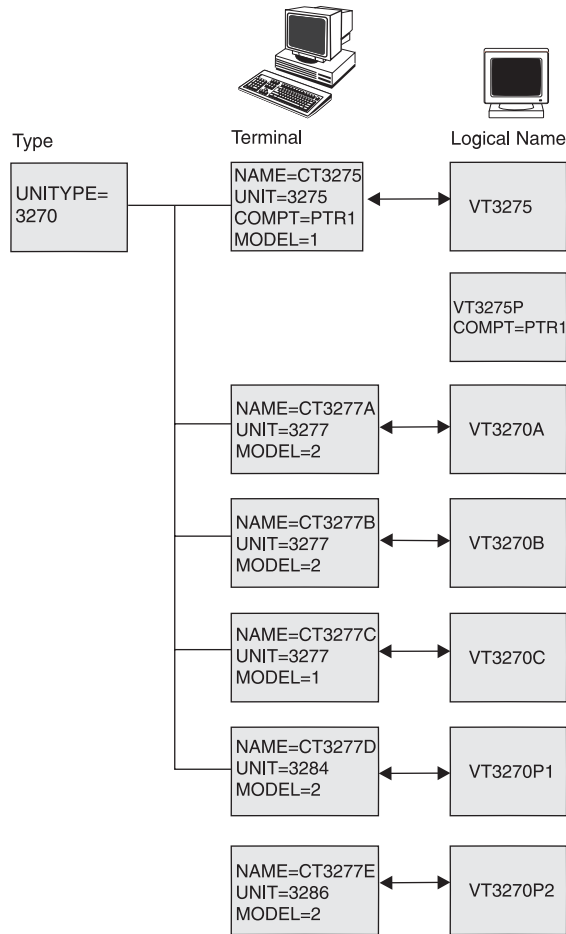


Figure 64. Remote 3270 VTAM Terminals

Figure 65 shows the macro statements to define remote 3270 VTAM terminals.

**Macro Statement Set Prepared**

```

TYPE UNITYPE=3270,MODEL=2,PTRSIZE=132,OPTIONS=COPY
TERMINAL NAME=CT3275,UNIT=3275,COMPT=PTR1,MODEL=1
NAME VT3275
NAME VT3275P,COMPT=PTR1
TERMINAL NAME=CT3277A
NAME VT3270A
TERMINAL NAME=CT3277B
NAME VT3270B
TERMINAL NAME=CT3277C,MODEL=1
NAME VT3270C
TERMINAL NAME=CT3277D,UNIT=3284
NAME VT3270P1
SPACE
TERMINAL NAME=CT3277E,UNIT=3286
NAME VT3270P2
    
```

Figure 65. Macro Statements to Define Remote 3270 VTAM Terminals

**Finance Communication System**

Figure 66 on page 504 illustrates macro statements for a Finance Communication System.

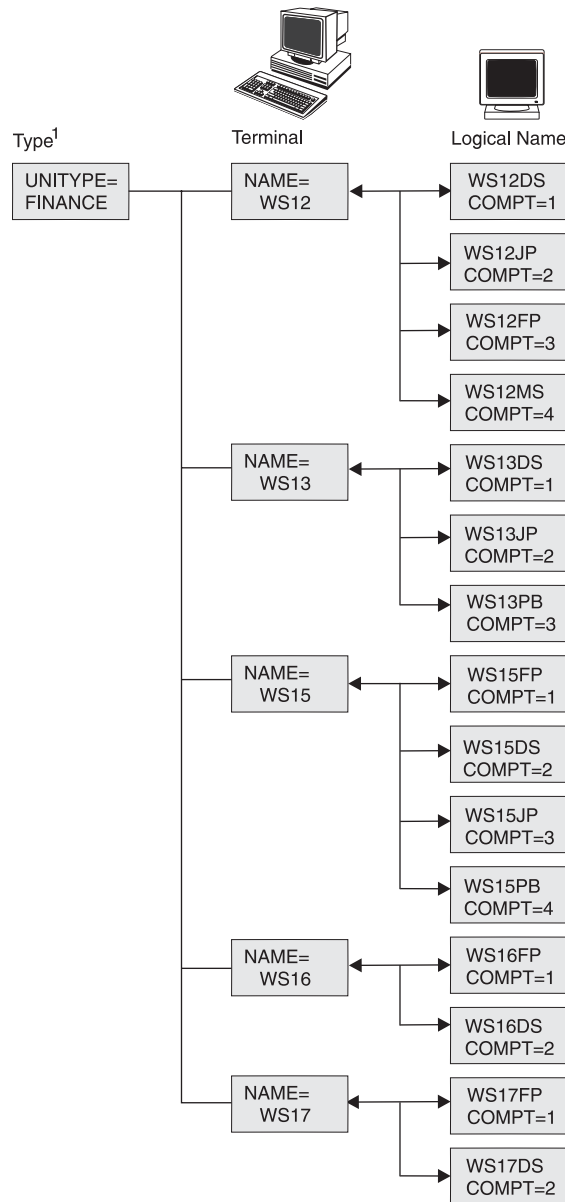


Figure 66. Finance Communication System. <sup>1</sup> Can also specify UNITYPE=3601.

Figure 67 on page 505 shows the macro statements to define a finance communication system.

Macro Statement Set Prepared	Column 72
TYPE UNITYPE=FINANCE,OUTBUF=256	
TERMINAL NAME=WS12,OPTIONS=MFS	
COMPT=(36DS3,36JP,36FP,36MS),FEAT=(DUAL,132)	
NAME WS12DS,COMPT=1	
NAME WS12JP,COMPT=2	
NAME WS12FP,COMPT=3	
NAME WS12MS,COMPT=4	
TERMINAL NAME=WS13,OPTIONS=(MFS,NOPNDST,OPTACK),	X
COMPT=(36DS4,36JP,36PB)	
NAME WS13DS,COMPT=1	
NAME WS13JP,COMPT=2	
NAME WS13PB,COMPT=3	
TERMINAL NAME=WS15,COMPT=(36FP,36DS,36JP,36PB),	X
OPTIONS=MFS	
NAME WS15FP,COMPT=1	
NAME WS15DS,COMPT=2	
NAME WS15JP,COMPT=3	
NAME WS15PB,COMPT=4	
TERMINAL NAME=WS16,COMPT=(36FP,36DS4),	X
OPTIONS=(FORCRESP,MFS,OPTACK,BID,OPNDST)	
NAME WS16FP,COMPT=1	
NAME WS16DS,COMPT=2	
TERMINAL NAME=WS17,COMPT=(36FP,36DS),	X
OPTIONS=(ACK,NOBID,TRANRESP,NOPNDST)	
NAME WS17FP,COMPT=1	
NAME WS17DS,COMPT=2	

Figure 67. Macro Statements to Define a Finance Communication System

### Secondary Logical Unit Type 1 VTAM Terminals

Figure 68 on page 506 illustrates the macro statements for a secondary logical unit type 1 VTAM terminal.

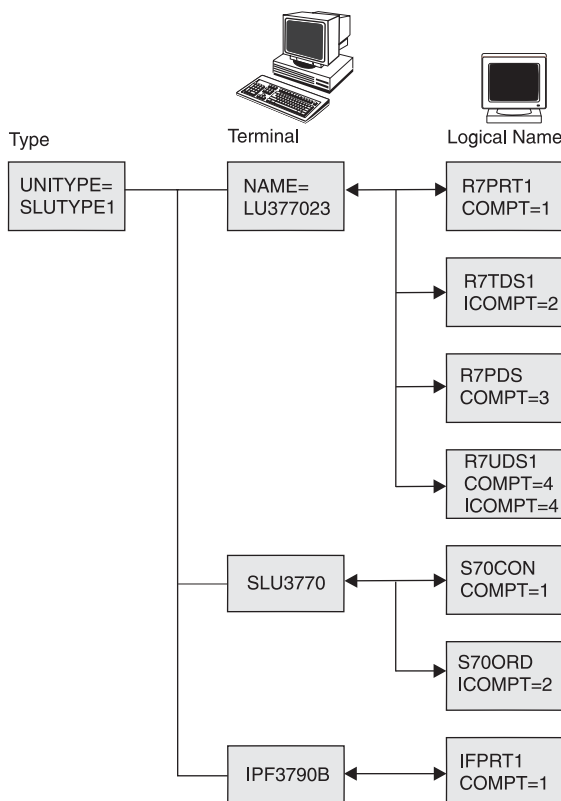


Figure 68. Secondary Logical Unit Type 1 VTAM Terminals

Figure 69 shows the macro statements to define SLU Type 1 VTAM terminals.

Macro Statement Set Prepared	Column
TYPE UNITYPE=SLUTYPE1	72
TERMINAL NAME=LU377023,	X
COMPT1=(PRINTER1,BASIC-SCS1),	X
COMPT2=(TRANSDS1,BASIC-SCS2),	X
COMPT3=(PRINTDS1,BASIC-SCS1),	X
COMPT4=(USERDS1,BASIC)	
NAME R7PRT1,COMPT=1	
NAME R7TDS1,ICOMPT=2	
NAME R7PDS1,COMPT=3	
NAME R7UDS1,COMPT=4,ICOMPT=4	
TERMINAL NAME=SLU3770,COMPT1=(CONSOLE,MFS-SCS1),	X
COMPT2=(READER1,MFS-SCS2,3)	
NAME S70CON,COMPT=1	
NAME S70RDR,ICOMPT=2	
TERMINAL NAME=IPF3790B,	X
COMPT1=(PRINTER1,BASIC-SCS1),	X
OPTIONS=(OPNDST,NBSELM)	
NAME IFPRT1,COMPT=1	

Figure 69. Macro Statements to Define SLU Type 1 VTAM Terminals

### NTO Devices

Figure 70 on page 507 illustrates macro statements for NTO devices.

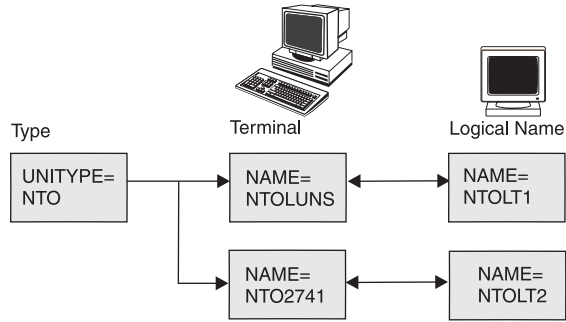


Figure 70. NTO Devices

Figure 71 shows the macro statements to define NTO devices.

Macro Statement Set Prepared	Column
	72
TYPE UNITYPE=NTO,EDIT=(EDIT1,EDIT2)	
TERMINAL NAME=NTOLUNS,EDIT=(YES,YES),	X
OUTBUF=300,SEGSIZE=300,PU=LUNS,OPTIONS=OPNDST	
NAME NTOLT1	
TERMINAL NAME=NTO2741,PU=2741,	X
MSGDEL=NONIOPCB,OPTIONS=(NORESP,MFS)	
NAME NTOLT2	

Figure 71. Macro Statements to Define NTO Devices

### Secondary Logical Unit Type 2 VTAM Terminals

Figure 72 illustrates macro statements for secondary logical unit Type 2 VTAM terminals.

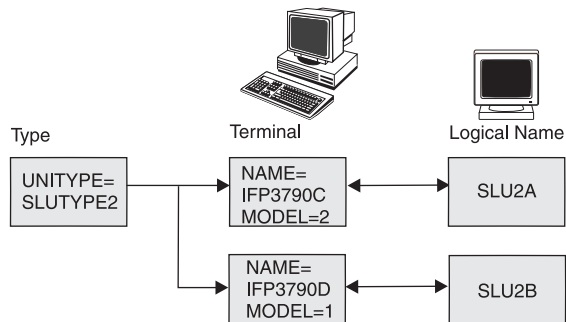


Figure 72. Secondary Logical Unit Type 2 VTAM Terminals

Figure 73 shows the macro statements to define SLU type 2 VTAM terminals.

```

Macro Statement Set Prepared

TYPE UNITYPE=SLU2TYPE2,MODETBL=AAA
TERMINAL NAME=IFP3790C,MODEL=2,MODETBL=BBB
NAME SLU2A
TERMINAL NAME=IFP3790D,MODEL=1,OUTBUF=500
NAME SLU2B
  
```

Figure 73. Macro Statements to Define SLU Type 2 VTAM Terminals

## Secondary Logical Unit Type P VTAM Terminals

Figure 74 illustrates the macro statements for secondary logical unit type P VTAM terminals.

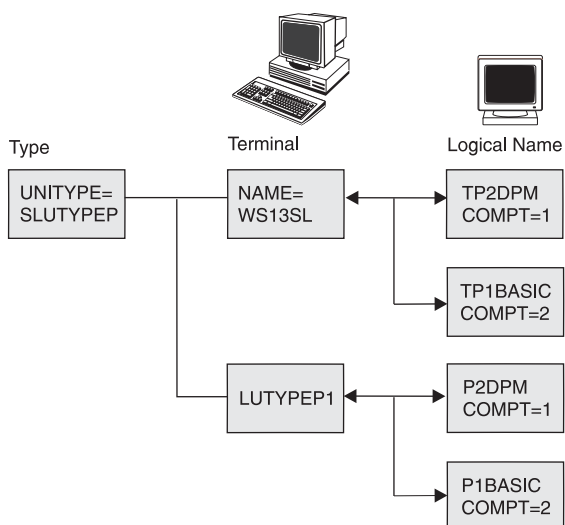


Figure 74. Secondary Logical Unit Type P VTAM Terminal

Figure 75 shows the macro statements to define SLU Type P VTAM terminals.

Macro Statement Set Prepared	Column
	72
TYPE UNITYPE=SLUTYPEP,OUTBUF=256, OPTIONS=(OPTACK,NOBID)	X
TERMINAL NAME=WS13SL, COMPT1=(PROGRAM2,DPM-A1,4), COMPT2=(PROGRAM1,BASIC)	X X
NAME TP2DPM,COMPT=1	
NAME TP1BASIC,COMPT=2	
TERMINAL NAME=LUTYPEP1, COMPT1=(PROGRAM2,DPM-A1,IGNORE), COMPT2=(PROGRAM1,DPM-A3,2), COMPT3=PROGRAM1	X X X
NAME P2DPM,COMPT=1	
NAME P1BASIC,COMPT=2	

Figure 75. Macro Statements to Define SLU Type P VTAM Terminals

## Logical Unit Type 6 VTAM Terminals

Figure 76 on page 509 shows the macro statements for Logical Unit Type 6 VTAM terminals.

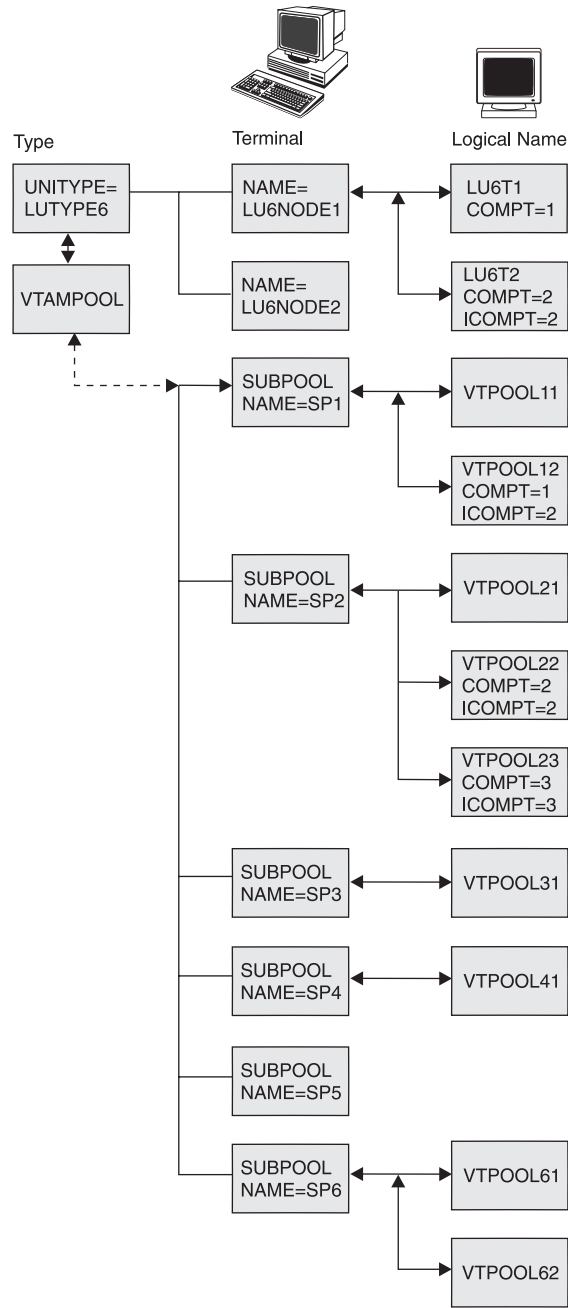


Figure 76. Logical Unit Type 6 VTAM Terminals

Figure 77 on page 510 shows the macro statements to define LU type 6 VTAM terminals.

Macro Statement Set Prepared	Column 72
TYPE UNITYPE=LUTYPE6	
*LU 6 NODE DEFINITION WITH FIXED LTERM ALLOCATION	
TERMINAL NAME=LU6NODE1,OPTIONS=NORESP,	X
COMPT1=(MULT2,DPM-B1,4),	X
COMPT2=(SINGLE2,VLVB)	
NAME LU6T1,COMPT=1	
NAME LU6T2,COMPT=2,ICOMPT=2	
*LU 6 NODE DEFINITION -- PARALLEL SESSIONS WITH DYNAMIC LTERM ALLOCATION	
TERMINAL NAME=LU6NODE2,OPTIONS=TRANRESP,	X
COMPT1=(SINGLE1,DPM-B1,IGNORE),	X
COMPT2=(SINGLE2,DPM-B3,2),	X
COMPT3=MULT1,	X
SESSION=3	
VTAMPOOL	
SUBPOOL NAME=SP1	
NAME VTPPOOL11	
NAME VTPPOOL12,COMPT=1,ICOMPT=2	
SUBPOOL NAME=SP2	
NAME VTPPOOL21	
NAME VTPPOOL22,COMPT=2,ICOMPT=2	
NAME VTPPOOL23,COMPT=3,ICOMPT=3	
SUBPOOL NAME=SP3	
NAME VTPPOOL31	
SUBPOOL NAME=SP4	
NAME VTPPOOL41	
SUBPOOL NAME=SP5	
SUBPOOL NAME=SP6	
NAME VTPPOOL61	
NAME VTPPOOL62	

Figure 77. Macro Statements to Define LU Type 6 VTAM Terminals

## System Configuration Macro Statement

In this example, the IMSGEN macro statement must be last in the IMS system definition input stream. The IMSGEN macro statement for that hypothetical system is shown in Figure 78 on page 511.



Characteristics to Define	Macro Statements Prepared	Column 72
IMS Configuration	IMSGEN ASM=HLASM, ASMPRT=ON, LKPRT=(XREF,LIST), LKSIZE=(200K,28K), LKRGN=200K, SUFFIX=0, OBJDSET=IMS.OBJDSET, USERLIB=IMS.SDFSRESL, PROCLIB=(YES,8), NODE=IMS, SCL=4, JCL=(SYSDEF, (A100,B20,M7044L21,'T=MAC')), MACKAY,0,(MSGLEVEL=(1,1), CLASS=N))	X X X X X X X X X X X X X X X

Figure 78. System Configuration Macro Statement

## IMS Multiple Systems Coupling

The Multiple Systems Coupling (MSC) sample shown in Figure 79 includes the relevant macros from the system definition of each of the three systems in a hypothetical multiple-IMS system configuration. The multiple-systems configuration is shown first, followed by extractions from the three system definitions in Figure 80 on page 512, Figure 81 on page 513, and Figure 82 on page 515. These system definitions do not include Intersystem Communication.

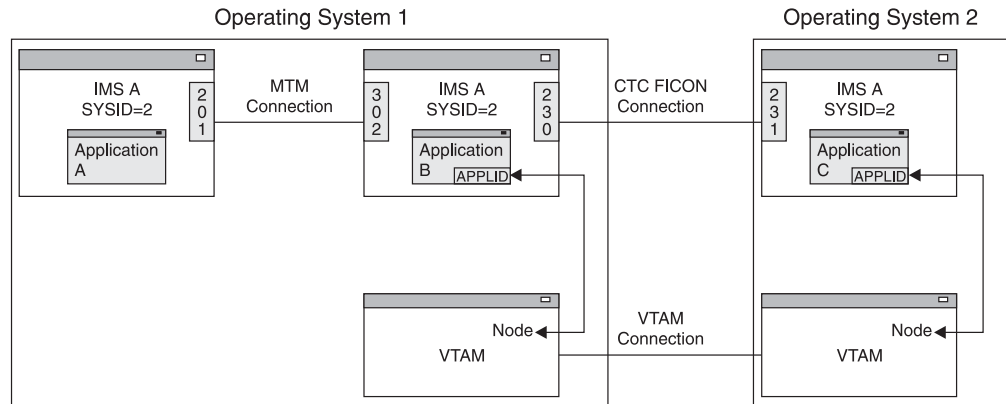


Figure 79. Multiple Systems Coupling

Characteristics to Define	Label	Macro Statements Prepared	Col. 72
IMS System Configuration		IMSCTRL MSVID=1,IMSID=SYSA IMSCTF	
Databases and Application Macros		DATABASE	
(Application programs that execute in		APPLCTN TRANSACTION APPLCTN PSB=A TRANSACTION CODE=A APPLCTN PSB=B,SYSID=(2,1)	

```

other systems          TRANSACT CODE=B
but process            APPLCTN PSB=C,SYSID=(3,1)
transactions          TRANSACT CODE=C
entered by
System A              APPLCTN
terminals are         TRANSACT
defined as
remote.)

Data                  COMM
Communication         LINEGRP
                      LINE
                      TERMINAL

(These logical
terminals are
defined in
Systems B and C
as remote
logical
terminals.)          LINEGRP
                      LINE
                      TERMINAL
                      NAME IMSATRM1
                      TERMINAL
                      NAME IMSATRM2
                      TERMINAL
                      NAME IMSATRM3

Multiple              LINK1 MSPLINK DDNAME=PLAB1,      X
Systems              ADDR=201,TYPE=CTC,              X
Coupling              BUFSIZE= 1000

                      LINK2 MSPLINK TYPE=MTM,          X
                      BUFSIZE=1000
                      MSLINK PARTNER=AB,              X
                      MSPLINK=LINK1

                      END

(Terminals in
Systems B and C
that can enter
transaction
code A are
defined as
remote
terminals.)          MSC12 MSNAME SYSID=(2,1)
                      NAME IMSBTRM1
                      NAME IMSBTRM2
                      NAME IMSBTRM3

                      MSC13 MSNAME SYSID=(3,1)
                      NAME IMSCTRM1
                      NAME IMSCTRM2
                      NAME IMSCTRM3
                      IMSGEN
    
```

Figure 80. System A IMS System Definition

Characteristics to Define	Label	Macro Statements Prepared	Col. 72
IMS System Configuration		IMSCTRL MSVID=2, IMSID=SYSB IMSCTF	
Databases and Applications  (Application programs that execute in other systems but process transactions entered by System B terminals are defined as remote.)		DATABASE APPLCTN TRANSACT APPLCTN PSB=B TRANSACT CODE=B APPLCTN PSB=A, SYSID=(1,2) TRANSACT CODE=A APPLCTN PSB=C, SYSID=(3,2) TRANSACT CODE=C  APPLCTN TRANSACT	
Data Communication  (These logical terminals are defined in Systems A and C as remote logical terminals.)		COMM LINEGRP LINE TERMINAL  LINEGRP LINE TERMINAL LINE LINEGRP LINE TERMINAL NAME IMSBTRM1 TERMINAL NAME IMSBTRM2 TERMINAL NAME IMSBTRM3 TERMINAL NAME IMSVBT1 TERMINAL NAME IMSVBT2 TERMINAL NAME IMSVBT3 TERMINAL NAME IMSVBT4	

Figure 81. System B IMS System Definition (Part 1 of 2)

```

Multiple
Systems
Coupling
LINK1  MSPLINK DDNAME=PLBA1,      X
        ADDR=302,TYPE=CTC,        X
        BUFSIZE=1000
LINK2  MSPLINK TYPE=MTM,          X
        BUFSIZE=1000
LINK3  MSPLINK DDNAME=PLBC1,      X
        ADDR=0BA,TYPE=BSC,        X
        BUFSIZE=360,              X
        CONTROL=YES
LINK4  MSPLINK TYPE=VTAM,          X
        NAME=NEWYORK,              X
        SESSION=2,                 X
        BUFSIZE=2048

        MSLINK PARTNER=AB,          X
        MSPLINK=LINK1
MSC21  MSNAME SYSID=(1,2)
        NAME IMSATRM1
        NAME IMSATRM2
        NAME IMSATRM3

        MSLINK PARTNER=BC,          X
        MSPLINK=LINK3
MSC23  MSNAME SYSID=(3,2)
        NAME IMSCTRM1
        NAME IMSCTRM2
        NAME IMSCTRM3

        MSLINK PARTNER=CB,          X
        MSPLINK=LINK4
MSC24  MSNAME SYSID=(4,6)
        NAME IMSVCT1
        NAME IMSVCT2

        MSLINK PARTNER=CC,          X
        MSPLINK=LINK4,             X
        OPTIONS=FORCESESS
MSC25  MSNAME SYSID=(5,7)
        NAME IMSVCT3
        NAME IMSVCT4

IMSGEN

END

```

Figure 81. System B IMS System Definition (Part 2 of 2)

Characteristics to Define	Label	Macro Statements Prepared	Col. 72
IMS System Configuration		IMSCTRL MSVID=3, IMSID=SYSC IMSCTF	
Databases and Applications		DATABASE	
(Application programs that execute in other systems but process transactions entered by System C terminals are defined as remote.)		APPLCTN TRANSACT APPLCTN PSB=C TRANSACT CODE=C APPLCTN PSB=A, SYSID=(1,3) TRANSACT CODE=A  APPLCTN PSB=B, SYSID=(2,3) TRANSACT CODE=B  APPLCTN TRANSACT	
Data Communication		COMM LINEGRP LINE TERMINAL	
(These logical terminals are defined in Systems A and B as remote logical terminals.)		LINEGRP LINE TERMINAL LINE LINEGRP LINE TERMINAL NAME IMSCTRM1 TERMINAL NAME IMSCTRM2 TERMINAL NAME IMSCTRM3 TERMINAL NAME IMSVCT1 TERMINAL NAME IMSVCT2 TERMINAL NAME IMSVCT3 TERMINAL NAME IMSVCT4	

Figure 82. System C IMS System Definition (Part 1 of 2)

```

Multiple      LINK1  MSPLINK DDNAME=PLCB1,      X
Systems      ADDR=0BB,TYPE=BSC,      X
Coupling     BUFSIZE=360,CONTROL=NO
             LINK4  MSPLINK TYPE=VTAM,      X
             NAME=HARTFORD,  X
             SESSION=2,      X
             BUFSIZE=2048
             MSLINK PARTNER=BC,      X
             MSPLINK=LINK1
MSC31        MSNAME SYSID=(1,3)
             NAME IMSATRM1
             NAME IMSATRM2
             NAME IMSATRM3
MSC32        MSNAME SYSID=(2,3)
             NAME IMSBTRM1
             NAME IMSBTRM2
             NAME IMSBTRM3
             MSLINK PARTNER=CB,      X
             MSPLINK=LINK4
MSC34        MSNAME SYSID=(6,4)
             NAME IMSVBT1
             NAME IMSVBT2
             MSLINK PARTNER=CC,      X
             MSPLINK=LINK4,      X
             OPTIONS=FORCSESS
MSC35        MSNAME SYSID=(7,5)
             NAME IMSVBT3
             NAME IMSVBT4
             MSGEN
             END

```

Figure 82. System C IMS System Definition (Part 2 of 2)

---

## Data-Sharing System Configuration

| These examples show four possible data-sharing system configurations. The IMS  
 | system definition macros that must be defined to support these configurations and  
 | the associated PSB definition statements are shown for each configuration. The  
 | illustrations in these examples assume that the database is registered with DBRC.

### Example 1. Data Sharing at the Database-Level with Update Access

Figure 83 on page 517 shows a possible configuration for data sharing.

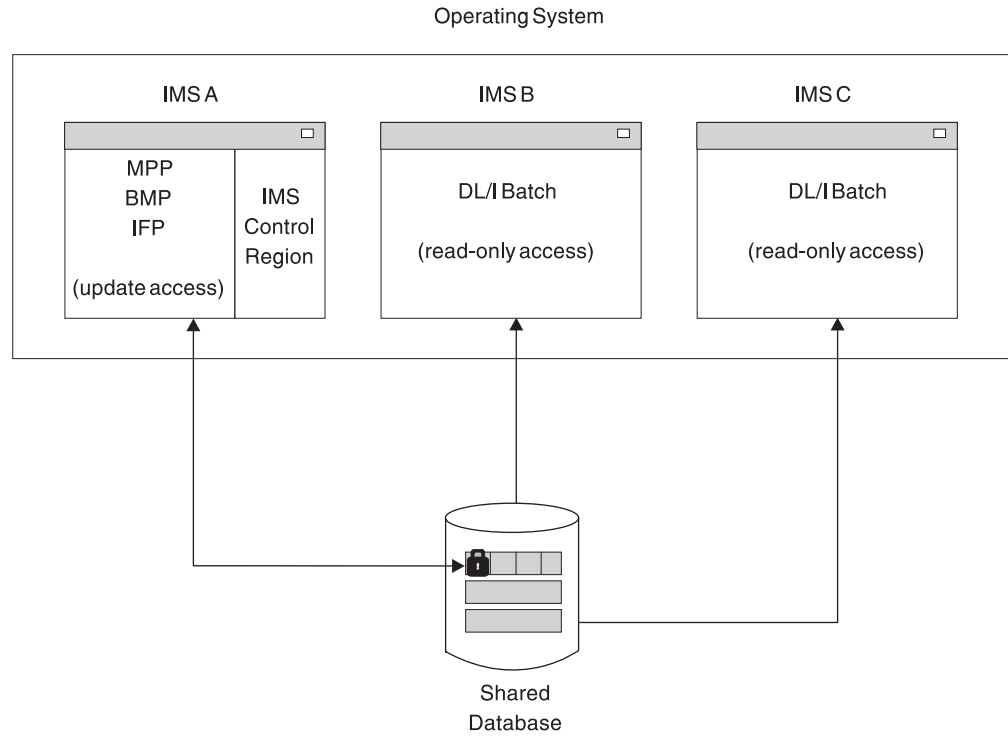


Figure 83. Data Sharing at the Database Level with Update Access

System A

IMS system definition statements include:

```
IMSCTRL  IMSID=IMSA,DBRC=YES
DATABASE DBD=SHRDB,ACCESS=UP
```

PSB definition statements include:

```
PCB      TYPE=DB,DBDNAME=SHRDB,PROCOPT=A
```

System B

IMS system definition statements include:

```
IMSCTRL  IMSID=IMSB,DBRC=YES,SYSTEM=(VS2,BATCH)
```

PSB definition statements include:

```
PCB      TYPE=DB,DBDNAME=SHRDB,PROCOPT=GO
```

System C

IMS system definition statements include:

```
IMSCTRL  IMSID=IMSC,DBRC=YES,SYSTEM=(VS2,BATCH)
```

PSB definition statements include:

```
PCB      TYPE=DB,DBDNAME=SHRDB,PROCOPT=GO
```

**Example 2. Data Sharing at the Database Level with Read Access**

Figure 84 on page 518 shows a possible configuration for data sharing.

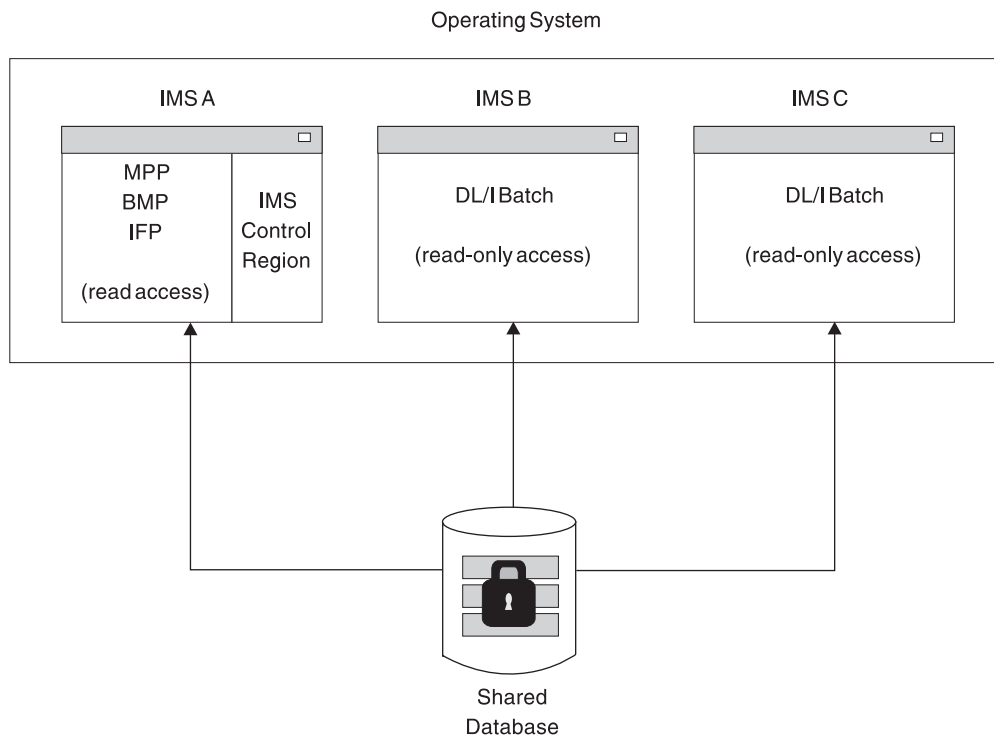


Figure 84. Data sharing at the Database Level with Read Access

#### System A

IMS system definition statements include:

```
IMSCTRL  IMSID=IMSA,DBRC=YES
DATABASE DBD=SHRDB,ACCESS=RD
```

PSB definition statements include:

```
PCB      TYPE=DB,DBDNAME=SHRDB,PROCOPT=G
```

#### System B

IMS system definition statements include:

```
IMSCTRL  IMSID=IMSB,DBRC=YES,SYSTEM=(VS2,BATCH)
```

PSB definition statements include:

```
PCB      TYPE=DB,DBDNAME=SHRDB,PROCOPT=GO
```

#### System C

IMS system definition statements include:

```
IMSCTRL  IMSID=IMSC,DBRC=YES,SYSTEM=(VS2,BATCH)
```

PSB definition statements include:

```
PCB      TYPE=DB,DBDNAME=SHRDB,PROCOPT=G
```

## Example 3. Intra-CPC Block-Level Data Sharing

Figure 85 on page 519 shows a possible configuration for data sharing.



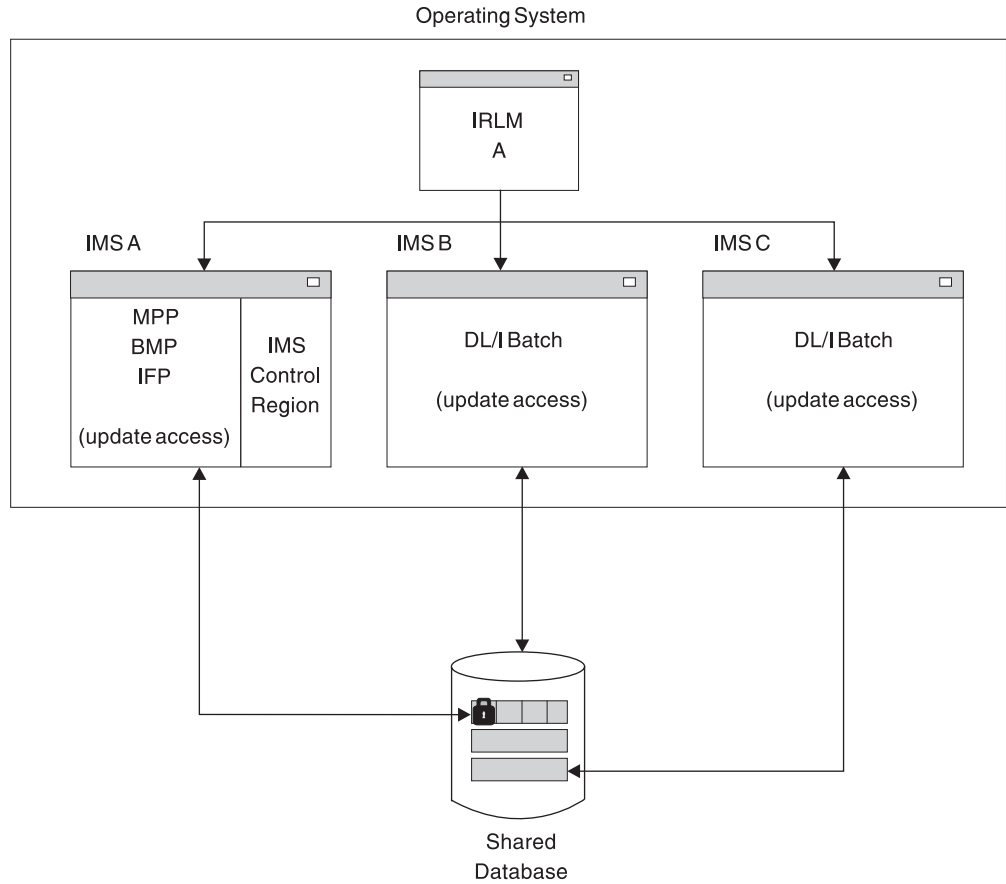


Figure 85. Intra-CPC Block-level Data Sharing

System A

IMS system definition statements include:

```
IMSCTRL  IMSID=IMSA,DBRC=YES,IRLMNM=RLMA
DATABASE DBD=SHRDB,ACCESS=RD
```

PSB definition statements include:

- PCB       TYPE=DB,DBDNAME=SHRDB,PROCOPT=A

System B

IMS system definition statements include:

```
IMSCTRL  IMSID=IMSB,DBRC=YES,IRLMNM=RLMA,       X
          SYSTEM=(VS2,BATCH)
```

PSB definition statements include:

```
PCB       TYPE=DB,DBDNAME=SHRDB,PROCOPT=GO
```

System C

IMS system definition statements include:

```
IMSCTRL  IMSID=IMSC,DBRC=YES,SYSTEM=(VS2,BATCH)
```

PSB definition statements include:

```

IMSCTRL  IMSID=IMSC,DBRC=YES,IRLMNM=RLMA,      X
          SYSTEM=(VS2,BATCH)
PCB      TYPE=DB,DBDNAME=SHRDB,PROCOPT=A
    
```

### Example 4. Inter-CPC Block-Level Data Sharing

Figure 86 shows a possible configuration for data sharing.

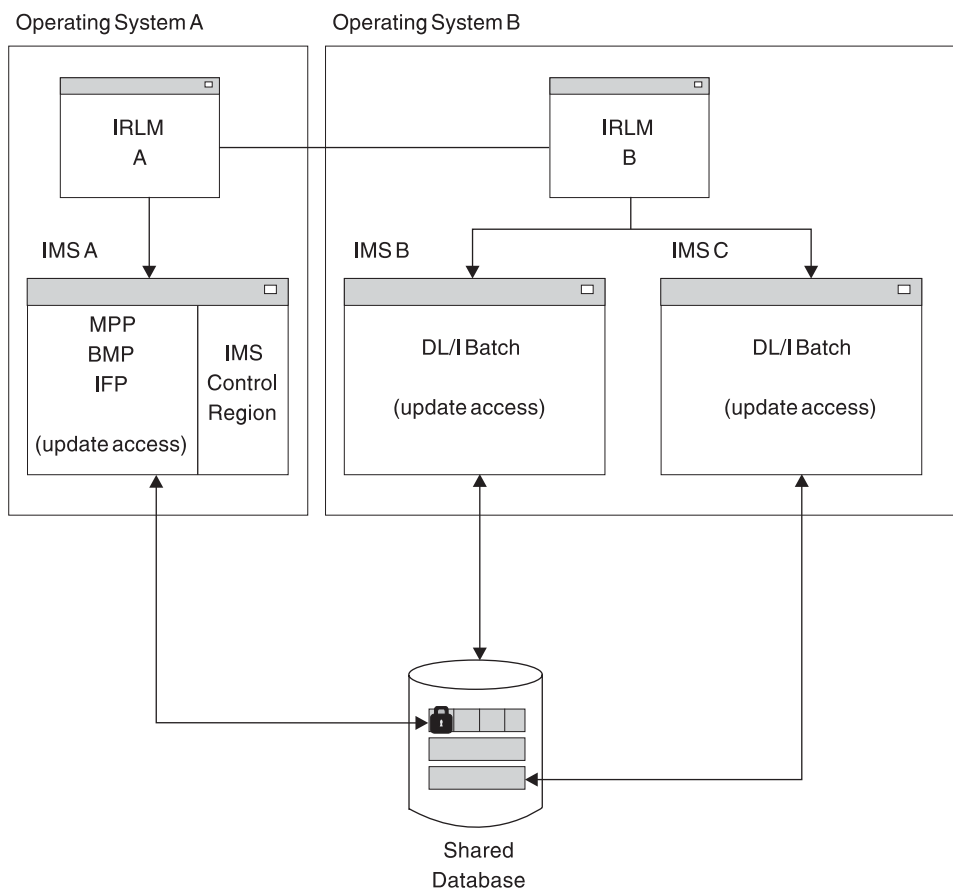


Figure 86. Inter-CPC Block-level Data Sharing

#### System A

IMS system definition statements include:

```

IMSCTRL  IMSID=IMSA,DBRC=YES,IRLMNM=RLMA
DATABASE DBD=SHRDB,ACCESS=UP
    
```

PSB definition statements include:

- PCB TYPE=DB,DBDNAME=SHRDB,PROCOPT=A

#### System B

IMS system definition statements include:

```

IMSCTRL  IMSID=IMSB,DBRC=YES,IRLMNM=RLMB,      X
DATABASE DBD=SHRDB,ACCESS=UP
    
```

PSB definition statements include:

```
PCB          TYPE=DB,DBDNAME=SHRDB,PROCOPT=A
```

### System C

IMS system definition statements include:

```
IMSCTRL     IMSID=IMSC,DBRC=YES,IRLMNM=RLMB,      X
SYSTEM=(VS2,BATCH)
```

PSB definition statements include:

```
PCB          TYPE=DB,DBDNAME=SHRDB,PROCOPT=A
PCB          TYPE=DB,DBDNAME=SHRDB,PROCOPT=A
```

---

## IMS DBCTL Environment

The following shows sample stage 1 input specifications for the DBCTL environment:

	Column 72
IMSCTRL SYSTEM=(VS/2,(ON-LINE,DBCTL),390),	X
MAXREGN=(20,52K,A,A),	X
MCS=(2,7),DESC=7,MAXCLAS=1,IMSID=IMSA,	X
CMDCHAR=4	
*	
IMSCTF SVCNO=(,203,202),	X
LOG=(DUAL,MONITOR),	X
RDS=(3380,4096),	X
CPLOG=1000	
*	
DEFINE SYSTEM BUFFERS	
*	
BUFPOOLS PSBW=60000,DMB=10000,	X
SASPSB=(20000,80000)	
*	
DEFINE FAST PATH OPTIONS	
FPCTRL OTHREAD=10,BFALLOC=(5,20,1024)	
*	
DEFINE SECURITY OPTIONS	
SECURITY TYPE=AGNEXIT	
*	
DEFINE DL/I DATABASES	
*	
DATABASE RESIDENT,DBD=DI21PART	
*	
DEFINE SAMPLE APPLICATIONS	
*	
APPLCTN PSB=DFHSAM04,SCHDTYP=PARALLEL	
APPLCTN PSB=DFHSAM05,SCHDTYP=PARALLEL	
*	
IMSGEN ASM=(HLASM,SYSLIN),	X
ASMPRT=ON,	X
LKPRT=(XREF,LIST),	X
LKSIZE=(880K,64K),	X
LKRGN=4096K,	X
SUFFIX=1,	X
SURVEY=NO,	X
SYMSG=TIMESTAMP,	X
OBJDSET=IMS.OBJDSET,	X
USERLIB=IMS.SDFSRESL,	X
PROCLIB=(YES,),	X
NODE=(IMS,IMS,IMS),	X
JCL=(GENJOB,	X
(1),	X
PGMERID,	X
A,	X
(TIME=5,CLASS=K,NOTIFY=PGMERID)),	X
SCL=(99)	
END	

---

## Appendix B. MFS System Definition and Programming Considerations

This appendix describes MFS-related system definition and programming considerations. For a complete description of IMS system definition macros, see Chapter 3, "Macros," on page 55.

**Related Reading:** For additional information on installation requirements, see *IMS Version 9: Installation Volume 1: Installation Verification*.

---

### System Definition Macro Instructions

The system definition macro instructions that include specifications for MFS are summarized in this section.

#### BUFPOOLS Macro

BUFPOOLS is used to define

- The size of the message format buffer pool
- The number of fetch request elements to be used for loading MFS control blocks into the message format buffer pool (with the FORMAT=, COMM=, and FRE= keywords).

**Related Reading:** For formulas used in calculating the buffer pool sizes, see *IMS Version 9: Administration Guide: System*. Default values are calculated and explained in Chapter 3, "Macros," on page 55. The pool size values set at system definition are used unless other values are specified at initialization of the online system control region.

The following I/O pool requirements for MFS are provided by the MFS language utility using message DFS1060I:

- Largest line buffer size for 3270 displays
- Largest work buffer size for an input message
- Largest work buffer size for an output message

For efficient operation in the virtual control region, the areas acquired for buffer pools during initialization can be fixed in address space during initialization.

#### MSGQUEUE Macro

MSGQUEUE is used to specify the maximum size output segment that will be properly processed by MFS.

#### IMSGEN Macro

IMSGEN is used to specify MFSTEST parameters if the COMM macro is not used. The SUFFIX= operand specifies the suffix of the MFS device characteristics table that is generated when TYPE or TERMINAL macros define device symbolic names (that is, TYPE=3270-An) during system definition.

#### COMM Macro

COMM is used to specify the following facilities of MFS:

- Exits for user-written field and segment editing routines to be included in the generated system

- IMS-provided formatting for MFS on the 3270 or SLU 2 master terminal

The IMS-provided formatting for the master terminal is provided if `OPTIONS=(...,FMTMAST,...)` is specified and the 3270 or SLU 2 display used as the master terminal is a 24x80 screen defined as `TYPE=(3270,2)` or `3270-An` with `SIZE=(24,80)`.

The name used for the `EDITNAME` parameter cannot be the same as any MID name.

## TYPE, TERMINAL, and CONFIG Macros

`TYPE`, `TERMINAL`, and `CONFIG` are used to describe terminal features.

The terminals that can be defined to operate with MFS by their number or by secondary logical unit type are described in Chapter 1, "Designing the IMS System," on page 1.

### TYPE Macro

The `TYPE`, `SIZE`, `FEAT`, and `OPTIONS` keywords can be used in the `TYPE` and `TERMINAL` macro. The `OPTIONS=` operand of the `TYPE` macro is used to request MFS support for a set of terminal devices described in the subsequent `TERMINAL` macros. If 3270 or SLU 2 terminals are defined, MFS support is automatic. The `TYPE`, `SIZE`, and `FEAT` keywords define the 3270 and SLU 2 devices using the device symbolic name and generate the MFS device characteristics table containing the device type symbolic name, associated screen size, and physical terminal features. `TYPE` keywords provide definition for subsequent `TERMINAL` macros.

For NTO, specify `UNITYPE=NTO` on the `TYPE` macro.

### TERMINAL Macro

The `COMPTn=` operand can be used to request MFS support for a specific component of a terminal defined as a secondary logical unit.

For NTO, specify `PU=xx` where `xx` is `LUNS`, `2741`, `2740-1`, or `TTY`.

The `COMPTn=` operand of the `TERMINAL` macro can request MFS support for the secondary logical units as follows:

**For `TYPE UNITYPE=SLUTYPE1`:** `COMPTn=(...,MFS-SCS1,...)` must be specified for the MFS SCS1 formatting option for the console, printer, or print data set component.

`COMPTn=(...,MFS-SCS2,...)` must be specified for the MFS SCS2 formatting option for the reader, punch, or transmit data set component.

**For `TYPE UNITYPE=SLUTYPEP`:** `COMPTn=(...,DPM-An,...)` must be specified for the MFS DPM formatting option for the remote controller component.

`COMPTn=(...,MFS-SCS1,...)` must be specified for the MFS SCS1 formatting option for the remote controller component.

**For `TYPE UNITYPE=LUTYPE6`:** `COMPTn=(...,DPM-Bn,...)` must be specified for MFS ISC formatting.

**Related Reading:** See the *IMS Version 9: Customization Guide* for MFS definition examples for ISC.

The following is a sample definition that would allow DPM-An to be used with a remote program. The definition is for a secondary logical unit with two components.

```

TYPE          UNITYPE=SLUTYPEP

TERMINAL     NAME=LUX,OPTIONS=(FORCRESP,ACK,NOBID),
              OUTBUF=256,COMPT1=(PROGRAM2,DPM-A3,7),
              COMPT2=(PROGRAM1,DPM-A4,1)

NAME         LTERMQ,COMPT=1

NAME         LTERMY,COMPT=2

```

PROGRAM1 is printed output operation; PROGRAM2 is paged output operation.

The OUTBUF= specification would allow RCDCTL to be up to 256 bytes in all format definitions for this secondary logical unit.

The following is a sample definition that would allow DPM-An or SCS1 to be used with a remote program. For this secondary logical unit, the DPM-An definition is for the first two components, and the SCS1 definition is for the third component.

```

TYPE          UNITYPE=SLUTYPEP

TERMINAL     NAME=LUX,OPTIONS=(FORCRESP,ACK,NOBID),
              OUTBUF=256,COMPT1=(PROGRAM2,DPM-A3,7),
              COMPT2=(PROGRAM1,DPM-A4,1),
              COMPT3=(PROGRAM1,MFS-SCS1,IGNORE)

NAME         LTERMQ,COMPT=1

NAME         LTERMY,COMPT=2

NAME         LTERMS,COMPT=3

```

PROGRAM1 is an unprotected operation at the end of a message for paged output or a printer (SCS1) operation; PROGRAM2 is a protected operation at the end of a message for paged output.

The OUTBUF= specification would allow RCDCTL to be up to 256 bytes in all DPM format definitions for this secondary logical unit.

If OUTBUF is not large enough for an entire data page, then more than one VTAM SEND is required to send the page.

**For TYPE UNITYPE=NTO:** OPTIONS=(...,MFS,...) must be specified on the TERMINAL macro for the MFS SCS1 formatting option to be used by the NTO device.

The following is a sample definition that would allow MFS to be used by an NTO device. MFS uses the SCS1 formatting option and treats the NTO device as an SLU 1 console. In this example, the NTO device is an IBM 3101 (LUNS).

```

TYPE          UNITYPE=NTO
TERMINAL     NAME=LU3101,PU=LUNS,
              OPTIONS=(TRANRESP,MFS,OPNDST),SEGSIZE=300
NAME         A3101

```

The TYPE, SIZE, FEAT, and OPTIONS keywords described in the TYPE macro example can be used in the TERMINAL macro.

The `TERMINAL` or `TYPE` macro indicates to MFS whether or not operator input that invokes processing for a new transaction should dequeue the output message for the current transaction. The parameters that apply are `OPTIONS=PAGDEL` or `NPAGDEL` or `PAGDEL=YES` or `NO`. Refer to “`TERMINAL Macro`” on page 143 for a discussion of the effects of these specifications.

### **CONFIG Macro**

The `CONFIG` macro describes the configuration for a switched 3275 terminal.

---

## **System Programming Considerations**

The major system programming considerations in implementing MFS are summarized in this section.

## **INDEX Directory**

The `INDEX` function of the MFS service utility creates an index directory to specified control blocks in the staging library, `IMS.FORMAT`. The IMS initialization process uses the index directory to construct the MFS dynamic directory in a private storage area. This space is separate from the message format buffer pool.

If a requested control block is not found in the message format buffer pool, MFS looks for an entry in the MFS dynamic directory. If found, the entry gives the relative track address for the indexed control block. This allows MFS to issue a direct read without going through the active format library's PDS directory. If the entry is not found, MFS reads the PDS directory, then reads the requested control block. MFS also creates an entry in the MFS dynamic directory for the control block.

The number of entries in the MFS dynamic directory continually increases until the MTO uses the command `/CHANGE DIRECTORY MFS`. This purges all the entries that were added to the MFS dynamic directory after online initialization.

Error conditions include:

- IMS abending during initialization if storage is unavailable for the MFS dynamic directory
- The `/MODIFY COMMIT` command failing during online change if storage is unavailable for the MFS dynamic directory or the PDS directory index
- No entries being added to the MFS dynamic directory during online operation if storage is unavailable for the MFS dynamic directory. To add entries to the MFS dynamic directory again, issue a `/CHANGE DIRECTORY MFS` command, or restart IMS.

## **FORMAT, FORMATA, FORMATB, and TFORMAT Library Directories**

Each PDS directory entry in the format libraries contains 14 bytes if created by a release of IMS prior to Release 1.6, or 24 bytes if created by IMS Release 1.6 or later. The 14-byte entry contains 12 bytes for the block name and TTR, and 2 bytes for the block size. The 24-byte entry adds 10 bytes for a data and time stamp and expansion of the block size to 4 bytes. IMS is able to read a directory containing a mixture of 14-byte and 24-byte entries.

## **Use of Concatenated Format Libraries**

You can define several format libraries on different physical or logical devices, thus permitting faster (concurrent) input/output processing. IMS limits the number of data sets that can be concatenated for MFS format libraries, `IMS.FORMATA` and `IMS.FORMATB` to 16. All of the concatenations must have like attributes.



Phase 2 of the MFS language utility implicitly deletes an old control block from a format library when a new control block with the same name is defined for that library. However, if the new control block is defined for a different library than the one the old copy is in, there will be no implicit deletion. Thus, old and new copies for control blocks can exist simultaneously in different format libraries.

One way to prevent this duplication is to explicitly delete duplicate control blocks, using the MFS service utility SCRATCH function. Another way is to add a DD statement in the MFSUTIL phase 2 JCL for each format library (up to 15) that is to have old control blocks deleted when new versions of those control blocks are defined for the staging library, IMS.FORMAT. Phase 2 will make the deletions, and will update the index directory \$\$IMSDIR (if there is one) in each library to reflect the new contents.

The new DD statements must have labels of the form, FORMATn, where n is a number from 1 to 15, inclusive. Phase 2 will only compress and add control blocks to the format library defined with a DD statement labeled FORMAT. (FORMAT is the original staging library.)

At IMS online initialization, the index directories (\$\$IMSDIR) of each of the format libraries are merged into a single MFS dynamic directory for all the format libraries. If a requested control block is not listed in the MFS dynamic directory, MFS will read one PDS directory block from each concatenated library, in sequence, until it finds the control block. This can mean many additional reads if the control block resides on one of the last concatenated libraries.

## Use of the Message Format Buffer pool

MFS control blocks are loaded into the message format buffer pool by one of two MFS pool manager routines: an immediate fetch routine or a prefetch routine. The immediate fetch routine loads a control block that is needed immediately and not already in storage. The prefetch routine is used only if the prefetch option is specified during initialization of the online IMS control region. It is invoked when IMS anticipates the need for a control block; if the block is already in storage, use of prefetch lowers the likelihood that the block will be forced out to make room for another block. The use of prefetch requests can improve response time.

When a requested control block is not already in the pool, and its name resides in the resident directory, \$\$IMSDIR, a single read operation loads the control block into storage. If the name is not in the resident directory, a read of one data set directory block followed by a read of the control block loads the control block into storage. Multiple reads of the data set occur when the size of the control block is greater than either the track size or the z/OS-specified BLKSIZE.

### Fetch Request Elements

Fetch request elements (FREs) are used to represent control blocks that are either already in storage or have been requested for loading into storage. The number of FREs to be defined depends on the number of control blocks that can reside in the pool (a function of pool size) and the number of concurrent requests to load control blocks. FREs defined during IMS system definition or during initialization of the online IMS control region are allocated contiguously in the buffer pool.

### Pool Space Management

The main objectives of message format buffer pool space management are:

- To keep as many control blocks as possible available in the pool
- To minimize pool fragmentation

The MFS pool manager assigns a priority to each fetch request. FREs represent prefetch requests and immediate fetch requests. When required, additional FREs are assigned from available pool space to represent immediate fetch requests. If an FRE is not available when a fetch request is made, a control block of lesser importance is discarded so that its FRE can be used to represent the new request.

When there is not enough pool space to hold a requested control block, a space-freeing algorithm is used to select less important control blocks (based on their frequency of use) for discarding.

### Pool Statistics

I Use the /DISPLAY P00L command to display the contents of the message format  
I buffer pool counters. The statistics displayed for directory and fetch input/output operations and pool space use should help in determining the appropriate pool size and number of FREs.

**Related Reading:** See *IMS Version 9: Administration Guide: System* for formulas to calculate the format buffer pool size and the number of FREs.

## Appendix C. Syntax Checker Example

The Syntax Checker is an ISPF application that helps you define, verify, and validate parameters and their values in the IMS.PROCLIB members. The syntax checker supports the following members for IMS Versions 7, 8, and 9:

- DFSPBxxx
- DFSDCxxx
- DFSSQxxx

The Syntax Checker provides an ISPF panel where you can add or change parameter values in the members. Online Help is available at the parameter level and provides assistance in moving to a new IMS release by identifying new parameters as well as any obsolete parameters from the previous release.

The Syntax Checker checks parameters for valid values and accepts only valid values. It saves the parameters to appropriate proclib members in the correct format. The next time you access the changed control region, it starts with the new values. There is no risk of syntax or parameter value errors.

The IMS IVP contains a Syntax Checker sample which presents a detailed example of using the Syntax Checker to migrate an IMS Version 8 DFSPBxxx proclib member to an IMS Version 9 level.

**Recommendation:** The IMS Syntax Checker does not perform security checking, nor does it bypass system security. You should protect your IMS data sets from unauthorized access by using RACF or an equivalent product.

### Function Keys and the Syntax Checker

Under ISPF, use the FKA ON command to display function keys, and use the KEYS command to set function keys. The Syntax Checker panels and Help panels have function keys assigned through an ISPF keylist. Although ISPF allows you to turn off the use of keylists, you should leave keylists on if you use the Help panels. Some function keys continue to be active, but are not shown when FKA ON is used. Table 59 describes the function keys used for the Syntax Checker.

*Table 59. Syntax Checker Function Key Descriptions*

Key	Description
F1=Help	Display Help information for panels and keywords.
F2=Split	ISPF split screen.
F3=Exit	Exit the current function. Returns back to the previous panel. If modifications have been made to the current member, the user is prompted to SAVE the member before returning back to the previous panel.
F6=Defaults	Display default values for keywords.
F7=Backward	Scroll backward. Pages backward through the list of parameters when the complete list of parameters cannot be displayed on one screen.
F8=Forward	Scroll forward. Pages forward through the list of parameters when the complete list of parameters cannot be displayed on one screen.
F9=Swap	ISPF swap screen command.
F10= Edit	Move cursor to Edit action bar.

Table 59. Syntax Checker Function Key Descriptions (continued)

Key	Description
F12=Cancel	Cancel action and return to previous panel.

## Starting the Syntax Checker

Start the Syntax Checker by issuing the following command in ISPF option 6:

```
EXEC 'HLQ.SDFSEXEC(DFSSCSRT)' 'h1q(HLQ)'
```

You can also start the Syntax Checker from the IMS Application Menu. The IMS Application Menu provides a common interface to IBM-supplied IMS applications that run on TSO and ISPF. The Syntax Checker remains running until you exit it manually.

**Related Reading:** For more information about the IMS Application Menu, see *IMS Version 9: Installation Volume 1: Installation Verification*.

## Syntax Checker Online Help

Help is available by pressing the Help key (usually the F1 key) or using the help pull-down menu whenever you are using the Syntax Checker.

- For any panel: Press the Help key to display a description of the panel.
- For the Syntax Checker Keyword Display Panel: To get specific keyword help, position the cursor anywhere on the keyword line and press the Help key

Help information is provided in a scrollable panel. The help information describes the Syntax Checker panel or selected keyword parameter.

## Using the Syntax Checker

After you start the Syntax Checker, the main panel (Syntax Checker Member and Data Set Name panel) shown in Figure 87 is displayed.

```

File Help
-----
                                IMS Parameter Syntax Checker
Command ===>

Enter the name of the IMS proclib dataset and press enter.

ISPF Library:
Project . . _____
Group . . . _____
Type . . . . _____
Member . . . _____ (Blank for member list)

Other Partitioned Data Set: 'IMS
Data Set Name . . 'HLQ.PROCLIB(DFSPBSYN)'
Volume Serial . . _____ (If not cataloged)

```

Figure 87. Syntax Checker Member and Data Set Name Panel

You use the Syntax Checker Member and Data Set Name panel to enter the IMS.PROCLIB data set name and the name of the member to be processed. To enter the names, use the standard TSO and ISPF formats:

- Enter the data set and member names using the ISPF Library: Project, Group, Type, and Member fields.
- Enter the data set and member name using the Other Partitioned Data Set: Data Set Name field.

If you do not enter the member name, a member list is displayed. Select the member you want to process from the list.

If the data set is not cataloged, enter the data set's volume in the Volume Serial field.

In this example, IMS PROCLIB data set name HLQ.PROCLIB (where HLQ is the high level qualifier) and member name DFSPBSYN are entered.

When you press Enter from the main panel, the Syntax Checker reads the input file and tries to determine the IMS release and type of control region. If the Syntax Checker cannot determine this information, one of the following panels displays:

- IMS Release and Control Region panel shown in Figure 88
- IMS Release panel shown in Figure 89

If the Syntax Checker is able to determine the information it requires from comments in the member, then the Syntax Checker Keyword Display panel shown in Figure 90 on page 533 displays.

## IMS Release and Control Region Panel

The IMS Release and Control Region panel, shown in Figure 88, provides the Syntax Checker with IMS release and control region information that is required to process the member correctly.

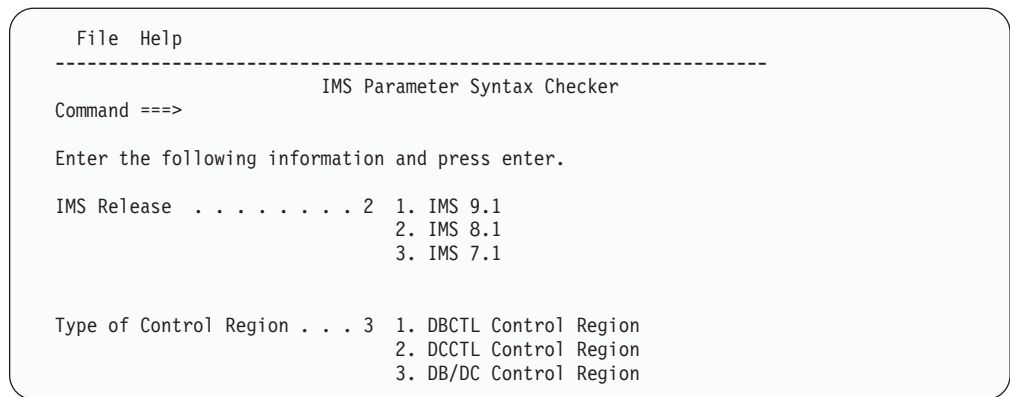


Figure 88. IMS Release and Control Region Panel

When the Syntax Checker saves the member, it adds comment lines to the top of the member saving this information. The next time the Syntax Checker processes the member, this panel does not display.

### Input Field Descriptions

The input fields for the IMS Release and Control Region panel include:

Field	Description
-------	-------------

**IMS Release** Enter 1, 2, or 3 to specify the IMS release.

This is a required field.

**Type of Control Region**

Enter 1, 2, or 3 to specify the type of control region.

This is a required field.

For this example, option 1 (IMS Version 9) is selected for the release and option 3 (DB/DC) is selected for the type of control region.

After you enter the data and press Enter, the Syntax Checker Keyword Display panel is displayed. "Syntax Checker Keyword Display Panel" shows the parameters that are currently specified for member DFSPBSYN.

## IMS Release Panel

The IMS Release panel, shown in Figure 89, provides the Syntax Checker with IMS Release information that is required to process the member correctly.

```

File Help
-----
                    IMS Syntax Checker
Command ==>

Enter the following information and press enter.

IMS Release . . . . . 1. IMS 9.1
                   2. IMS 8.1
                   3. IMS 7.1
  
```

*Figure 89. IMS Release Panel*

Enter 1, 2, or 3 to specify the correct IMS Release information. This is a required field.

After you enter the data and press Enter, the Syntax Checker Keyword Display panel is displayed. "Syntax Checker Keyword Display Panel" shows the parameters that are currently specified for member DFSPBSYN.

## Syntax Checker Keyword Display Panel

The Syntax Checker Keyword Display panel, shown in Figure 90 on page 533, is the main working panel. It displays the keywords and their values and indicates any errors.

```

File Edit View Help
-----
                    IMS 9.1 Parameters for DB/DC
Command ==>>>

Press enter (without other input) to check the syntax.

Data Set Name . . . : HLQ.PROCLIB(DFSPBSYN)
IMS Release . . . : 9.1

Sel Codes: C = Comment P = Process D = Delete / = Select'

Sel Keyword      Value      Description      More: +
-----
- APPLID = IMS1      Unknown Keyword
- XXXXX  = YYYYYY    Unknown Keyword
- ALOT   = 9          ETO Auto Logon Off Time
- AOIS   = A          ICMD Security Option
- APPC   = Y          Activate APPC/IMS (Y|N)
- APPLID2 = IMS2     VTAM Applid of XRF Alternate System
- ARC    = 99         Automatic Archive: 1-99, 0 - NOT Autom.
- ASOT   = 1000      ETO Auto Signoff Time
- AUTO   = Y          Automatic Restart Desired (Y|N)
- CHTS   = 1000      Number of CCB Hash Table Slots
- CMDMCS = N          MCS/EMCS Command Option: N|Y|R|C|B
- DBBF   = 1000      Number of Database Buffers
- DBFX   = 10         Num. DB Buffs available at FP Reg Start
- DBRCNM = DBCPROC   DBRC Proplib Member Name
  
```

Figure 90. Syntax Checker Keyword Display Panel

### Display Options

The Syntax Checker provides the following display options that you can select from the View pull-down menu:

- Select Display - Display all keywords with values. These keywords are saved when the member is saved.
- All Display - Display all possible keywords. Only keywords with values are saved when the member is saved.
- New Display - Display all new keywords in this release of IMS. These keywords are saved only if they have values when the member is saved.

The status of the member determines whether keywords are displayed the first time the panel is displayed:

- If the member is new or empty, then a list of all possible keywords for the member is displayed (All Display). The list of keywords is shown in alphabetical order.
- If the member is not empty or new, then the list contains the current keywords defined in the member (Select Display). The list is shown in alphabetical order by keyword name. You can obtain a display of all possible keywords for the member by using the View pull-down Display All option.

To display the default values for parameters, press the F6 function key to toggle between displaying and not displaying defaults. The default values are displayed in the description field of the parameter. When default values are displayed, the description or comment appears to the left of the field. Shifting the display to the right or left is not supported.

### Panel Field Descriptions

This section describes the fields on the Syntax Checker Keyword Display panel.

Field	Description
Panel Title	The IMS release and control region.
Data Set Name	The IMS.PROCLIB data set and member name.
IMS Release	The release of the IMS system to which the parameters apply.
Sel	<p>An input field that allows you to:</p> <ul style="list-style-type: none"> <li>• Select with a C to process the keyword using the Comments command.</li> <li>• Select with a D to process the keyword using the Delete command.</li> <li>• Select with a P to interrupt the processing of the current member and begin processing the member identified by the selected parameter. For more information about this option, see “Interrupting Processing to Process Other Members” on page 537.</li> <li>• Select with a / to process the keyword using the Action pull-down option.</li> </ul>
Keyword	<p>The name of the keyword. A blank keyword and value line is provided on the top line of the display to allow the entry of a new keyword and value.</p> <p>The keywords are color-highlighted as follows:</p> <p><b>Green</b> Normal color for a keyword with a value.</p> <p><b>Blue</b> A new keyword in this release.</p> <p><b>Yellow</b> Warning: Read the help text.</p> <p><b>Red</b> Keyword Error</p> <p><b>Turquoise</b> Keyword is a template. The keyword has no value and will not be saved. These keywords are only displayed in the DISPLAY ALL panel.</p>
Value	<p>The modifiable field which contains the value assigned to the keyword parameter. You can position the cursor on the input field and press the help key to get more information about the keyword parameter.</p> <p>The keywords are color-highlighted as follows:</p> <p><b>Turquoise</b> Value is correct.</p> <p><b>Red</b> Value error.</p>
Description/Comment	<p>Provides a short description of the keyword parameter or a user-provided comment. You can modify both the description and the comment. A modified description becomes a user comment. If the user comment is set to all blanks, the description is displayed. The user comment is written to the member when the member is saved, but the description provided by the Syntax Checker is not. The user-provided comment can be a maximum of 42 characters.</p>



Comments are not supported by all members. See “Specifying Comments” for information about specific members that support comments.

**Note:** For the DFSPBxxx members, an equal sign (=) is not allowed in a user-provided comment. It is changed by the Syntax Checker to a dash (-). Message DFSI937I Equal sign in same line comment changed to dash is displayed.

## Error Checking

After modifying the member, press the Enter key without making any other modifications. Syntax /value checking is performed. If there are errors, the first keyword with an error moves to the top of the display and an error message is displayed. The error must be corrected before the next error can be displayed.

For example, in Figure 90 on page 533, the Syntax Checker Keyword panel contains deliberate errors labeled as Unknown Keywords. On the screen, each error shows up in red on the screen and is displayed at the top of the list of keywords.

The first error (APPLID=IMS1) is a typographical error. Adding a 1 to the end of APPLID makes it a valid parameter (APPLID1).

The second error (Keyword XXXXX) is not a valid keyword. To resolve this error, delete the keyword. In this example, a “D” (delete) select code is typed into the Sel field next to the XXXXX.

The third error is that the AUTO parameter has an erroneous value of X instead of a correct value of Y or N. If you press the Enter key with no other input, the AUTO parameter is displayed on the top line of the display and an error message describing the error is displayed.

When all errors are resolved and you have modified the member as required, you can save the member to the originally selected PROCLIB and member or to a different PROCLIB and member.

## Specifying Comments

IMS allows comments to be included in most IMS.PROCLIB members. The comments can be either full-line comments or comments that appear on the same line as the parameters. The style of comment depends upon the specific member being processed.

In general, full-line comments are indicated by an asterisk (\*) or a slash-asterisk (/\*) in column 1. Each comment is associated with the parameter keyword that follows the comment. When the file is written, the comments precede the keyword. You can add, edit or delete comments in the following ways:

- Use the select option, C, to select the keyword associated with, or to be associated with, the comment. Press Enter.  
The panel redisplay with a blank comment line above the selected keyword. Add your comment to the blank line.
- The comment can be modified at any time.
- To delete a comment, enter the select option, D, next to the comment line. Press Enter.

Table 60 on page 536 describes the limitations that you should consider when specifying comments for specific members:

Table 60. Special Considerations for Comments in PROCLIB Members Processed by the Syntax Checker

Member	Considerations When Specifying Comments
DFSDCxxx	Comments must be specified on a separate line; comments are not allowed on the same line as parameters. You cannot edit the description of the parameter.
DFSSQxxx	Comments are not supported.
DFSPBxxx	Comments can be specified on a separate line, or they can be specified on the same line as parameters. A comment is separated from the parameter by one or more blanks. Comments can contain any characters except the equal sign (=).

## Action Pull-Down Options

Action pull-down options are located at the top of the panel. Selecting an Action pull-down option displays the options available for the action pull-down.

### File Action Pull-Down Options:

**Save** The keywords are stored back into the originally selected proclib member.

**Save As** The keywords are stored to a specific data set and member. You are prompted for the data set name and member name.

For more information, see "Saving Processed Members" on page 537

### **Change Release**

This option changes the IMS release of the parameter member being processed as follows:

1. If any changes have been made to the current member, you are given an option to save the member before changing the release.
2. The member is reprocessed under the new release, identifying any value errors and invalid or obsolete keywords.

### Edit Action Pull-Down Options:

**Comments** Adds a comment to keywords that have been selected with the / operand.

**Delete** Deletes from the display keywords that have been selected with the / operand.

**Delete All** Deletes the ALL keywords from the display.

### View Action Pull-Down Options:

**Display All** Changes the keyword display to include all possible keywords for the selected IMS release and control region type. The list is shown in alphabetical order, sorted by keyword name.

### **Display Selected**

Changes the keyword display to include only the keywords that have a value. The list is shown in alphabetical order, sorted by keyword name. This is the list of parameters that are saved when the save command is issued.

**Display New** Changes the keyword display to include only the keywords that are new for the IMS release. The list is shown in alphabetical order, sorted by keyword name.

### Interrupting Processing to Process Other Members

While using the Syntax Checker Keyword Display Panel, you can select the P option to interrupt processing of the current member and begin processing another member. The P option is valid on any keyword whose value is used to identify a member in the IMS.PROCLIB. The keyword displays as underlined for easy identification. The keyword must have a value.

The selected member is processed under the current IMS release. When processing of the selected member is complete (F3), processing of the interrupted member returns.

If the member does not exist, the panel shown in Figure 91 is displayed to ask if a new member should be created or if the member should be processed in a different data set. If neither action is appropriate, the user can choose to cancel processing of the member.

```

IMS Syntax Checker
Command ==>>

The Member DFSDC001 does not exist in dataset 'xxx.yyy.zzz'
Select one of the following options:

Option: __
1. Create a new member in current dataset
2. Cancel processing of the member
3. Process member in different dataset:
_____ (Dataset name)
    
```

Figure 91. Member Does Not Exist Panel

If the member is not one of the members processed by Syntax Checker, the panel shown in Figure 92 is displayed to ask if the user wants to edit (or create) the member in the current data set, edit (or create) the member in a different data set, or cancel processing of the member.

```

IMS Syntax Checker
Command ==>>

The Member DFSDC001 is not processed by Syntax Checker.
Select one of the following options:

Option: __
1. Edit/Create member in current dataset
2. Cancel processing of the member
3. Edit/Create member in different dataset:
_____ (Dataset name)
    
```

Figure 92. Member is Not Processed by Syntax Checker Panel

### Saving Processed Members

The processed member can be saved to the originally selected PROCLIB and member or to a different PROCLIB and member. To save the member, you use one of the Save or Save As options described in “Action Pull-Down Options” on page 536. The parameters are stored back into the member as follows:

- The parameters are stored in alphabetical order.
- Only the parameters that are displayed in the Display Selected keyword panel are stored. These parameters are either read in from the member or specifically assigned a value during Syntax Checker processing of the member.
- For members that support comments:
  - Comments are stored in front of their keyword.
  - Syntax Checker adds the following informational comments to the top of the member:
 

```
IMS Version:          *<version> 9.1
Last Modified date:  *<date> 2003/06/25
Last Modified time: *<time> 15:34
Modifying userid:   *<sysuid> GR8ONE
```

## Save Member Prompt Panel

The Save Member Prompt panel shown in Figure 93, notifies you of an issue or problem before the Syntax Checker saves the member or continues without saving the member. You are prompted to SAVE or NOT SAVE the member currently being processed.

```

File Edit View Help
- +-----+-----+
|          IMS Parameter Syntax Checker          |
| C Command ====>                               |
| P                                              |
| D Member DFSPBSYN has been modified.          |
| I Do you wish to save it?                     |
| Save Member ? . . . . . 2 1. YES              |
|                                     2. NO      |
+-----+-----+
- APPLID2 = IMS2          VTAM Applid of XRF Alternate System
- ARC      = 99          Automatic Archive: 1-99, 0 - NOT Autom.
- ASOT     = 1000       ETO Auto Signoff Time
- AUTO     = Y          Automatic Restart Desired (Y|N)
- CHTS     = 1000       Number of CCB Hash Table Slots
- CMDMCS   = N          MCS/EMCS Command Option: N|Y|R|C|B
- DBBF     = 1000       Number of Database Buffers
- DBFX     = 10         Num. DB Buffs available at FP Reg Start
- DBRCNM   = DBCPROC    DBRC Prohibit Member Name

```

Figure 93. Save Member Prompt Panel

The input field for this panel includes:

Save Member ?

- Enter 1 to save the member.
- Enter 2 to continue without saving the member.

This is a required field.

For more information about how parameters and comments are stored when you save a member, see “Saving Processed Members” on page 537.

## SAVE AS Prompt Panel

The SAVE AS Prompt panel, shown in Figure 94 on page 539 provides the Syntax Checker with the data set name and the member name in which to save the current

keywords being processed. Enter the data set name and member name in the standard TSO and ISPF formats.

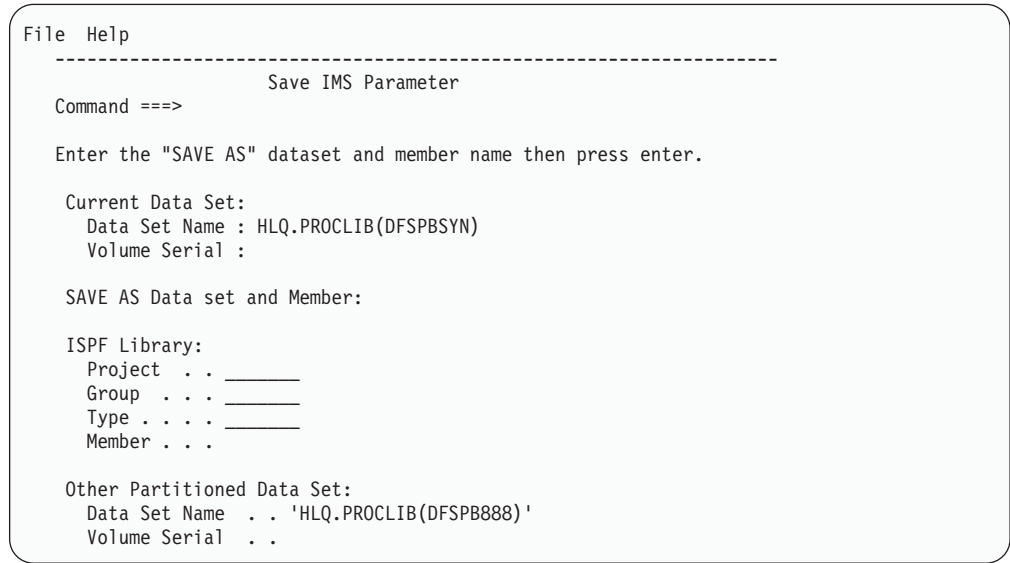


Figure 94. SAVE AS Prompt Panel

- Enter the data set and member names using the ISPF Library: Project, Group, Type, and Member fields.
- Enter the data set and member names using the Other Partitioned Data Set: Data Set Name field.

If the data set is not cataloged, enter the data set's volume in the Volume Serial field.

|  
|

For more information about how parameters and comments are stored when you save a member, see "Saving Processed Members" on page 537.

## Change Release Panel

The Change Release panel, shown in Figure 95 on page 540, provides the Syntax Checker with the new IMS Release to use in order to process the keywords.

The Change Release option can be found on the File pull-down menu.

Keywords are processed again under the new release to identify any value errors and invalid or obsolete keywords.

The input field for this panel includes:

Field	Description
<b>IMS Release</b>	Enter 1, 2, or 3 to specify the IMS release. This is a required field.

This is a required field.

```

IMS Parameter Syntax Checker
Command ==>>

Select New Release and press Enter.

IMS Release . . . 1  1. IMS 9.1
                   2. IMS 8.1
                   3. IMS 7.1

Sel Keyword      Value      Description          More: +
-  APPLID2      = IMS2      VTAM Applid of XRF Alternate System
-  ARC          = 99        Automatic Archive: 1-99, 0 - NOT Autom.
-  ASOT         = 1000     ETO Auto Signoff Time
-  AUTO         = Y        Automatic Restart Desired (Y|N)
-  CHTS         = 1000     Number of CCB Hash Table Slots
-  CMDMCS       = N        MCS/EMCS Command Option: N|Y|R|C|B
-  DBBF         = 1000     Number of Database Buffers
-  DBFX         = 10       Num. DB Buffs available at FP Reg Start
-  DBRCNM       = DBCPROC   DBRC Proclib Member Name

```

Figure 95. Change Release Panel

---

## Exiting the Syntax Checker

| To exit the Syntax Checker, press the exit key (usually, F3), until you return to ISPF.  
 | The Syntax Checker will prompt you to save any unsaved changes:

- | • Enter 1 to save the member.
- | • Enter 2 to continue without saving the member.

## Appendix D. IMS Connect Dump Formatter

This topic contains Diagnosis, Modification, or Tuning Information.

The IMS Connect Dump formatter enables you to format various IMS Connect internal control blocks under the ISPF IPCS environment. An IBM service support representative might ask you to use the IMS Offline Dump Formatter to diagnose an IMS Connect problem.

The IMS Offline Dump Formatter enables you to display the following IMS Connect control blocks:

- EST
- SCT
- DCT
- INTT
- SVT
- DST
- HCDB
- FWE
- TWU
- CTOKN

You can also format IMS Connect BPE control blocks and trace entries from within the IMS Connect Dump formatter. IMS Connect executes under the IMS Connect BPE.

**Prerequisite:** Before you install the IMS Connect portion of the Interactive Dump formatter, ensure that IPCS is already working with ISPF/PDF and the IMS formatting functions. First, update the following DD concatenations in Table 61:

*Table 61. DD Concatenations*

DDNAME	Data set to be added	Contents of data set
SYSPROC	CONN.SHWSCLIST	CLISTs
ISPLIB	CONN.SHWSPLIB	PANELs
ISPTLIB	CONN.SHWSTLIB	TABLEs
TASKLIB	IMS.RESLIB or CONN.RESLIB	Formatting modules (The TASKLIB concatenation is specified as part of the IPCS command invocation)

After you have updated the concatenations described in Table 61, ensure that the IMS Connect dump formatter modules are in a data set that is included in the IPCS TASKLIB data set concatenation. You can either:

- Include the IMS Connect dump formatter RESLIB data set in the IPCS TASKLIB concatenation
- Copy the IMS Connect dump formatter modules into a separate data set in the IPCS TASKLIB concatenation

The modules that must be copied from the IMS Connect RESLIB include:

- HWS\$B130
- HWS\$F130
- HWS\$S130

- HWSODF0\$
- HWSFT220
- HWSOB220

## Accessing the IMS Connect Dump Formatter

The way that you access the IMS Connect Dump Formatter depends on the version of IMS that you are using.

IMS Version 7 users can access the IMS Connect Dump formatter from the IMS Offline Dump Formatter ISPF panels.

The rest of this topic describes, by version of IMS, how to access the IMS Connect Dump Formatter.

### IMS Version 7 Users

If you have the IMS Version 7 Offline Dump Formatter installed, access the IMS Connect Dump Formatter as follows.

1. Select DFSAAMPR from the IPCS Component Analysis panel. You will see the panel in Figure 96 where the primary menu options for IMS Dump Formatting appears.

```

DFSAAMPR ----- IMS DUMP FORMATTING PRIMARY MENU -----
OPTION ==>

  0 INIT          - IMS formatting initialization and content summary
  1 BROWSE        - Browse Dump data set (IPCS norm)          *****
  2 HI-LEVEL      - IMS Component level formatting           *USERID - IMSDUMP
  3 LOW-LEVEL     - IMS ITASK level formatting                *DATE   - 00/08/23
  4 ANALYSIS     - IMS dump analysis                         *JULIAN - 00.236
  5 USER         - IMS user formatting routines              *TIME   - 16:06
  6 OTHER COMP   - Other IMS components (BPE, CQS...)        *PREFIX - IMSDUMP
  7 OTHER PROD   - Other IMS-related products                *TERMINAL- 3278
  E EDA         - IMS Enhanced Dump Analysis                *PF KEYS -
  T TUTORIAL     - IMS dump formatting tutorial              *****
  X EXIT        - Exit IMS dump formatting

Enter END or RETURN command to terminate IMS component formatting.

Use PFKeys to scroll up and down if needed.

```

Figure 96. IMS Dump Formatting Primary Menu Panel

2. Select option 7 Other Prod from the IMS dump formatting primary menu. After you select option 7, the IMS-Related Product Selection Dump Formatting Menu panel, as shown in Figure 97 on page 543, appears. The IMS-Related Product Selection Dump Formatting Menu lists options for browsing dump data set, formatting IMS Connect, and formatting IMS ORS.



```

----- IMS-RELATED PRODUCT SELECTION DUMP FORMATTING MENU -----
OPTION ==>

  1 BROWSE      - Browse dump data set (IPCS norm)      *****
  2 IMS Connect - IMS Connect formatting                *USERID - IMSDUMP
  3 ORS         - IMS ORS formatting                   *DATE   - 00/08/23
                                                    *JULIAN - 00.236
                                                    *TIME   - 16:08
                                                    *PREFIX - IMSDUMP
                                                    *TERMINAL- 3278
                                                    *PF KEYS - 24
                                                    *****

  X EXIT       - Exit IMS component menu

Enter END or RETURN command to terminate IMS Connect formatting.

The above selections are only valid if the indicated products are installed.

```

Figure 97. IMS-Related Product Selection Menu Panel

3. Select option 2 IMS Connect. The IMS Connect dump formatter menu appears (see Figure 98). “Using the IMS Connect Dump Formatter” describes how to use the options on the main IMS Connect dump formatter menu.

**Related Reading:** See the “Formatting IMS Dumps Offline” section of Chapter 7, “SYS — System Service Aids” in the *IMS Version 7 Diagnosis Guide and Reference* for an overview of the IMS Offline Dump Formatter and how to use it.

## Using the IMS Connect Dump Formatter

Figure 98 shows the IMS Connect Dump Formatting Menu panel. The IMS Connect Dump Formatting Menu lists four options: show BPE status and initialize the dump; browse the dump data set; hi-level IMS Connect component level formatting; and low-level IMS Connect level formatting.

```

----- IMS CONNECT DUMP FORMATTING MENU -----
OPTION ==>

  0 INIT       - Show BPE status and initialize dump
  1 BROWSE     - Browse dump data set (IPCS norm)      *****
  2 HI-LEVEL   - IMS Connect component level formatting *USERID - IMSDUMP
  3 LOW-LEVEL  - IMS Connect level formatting          *DATE   - 00/08/23
                                                    *JULIAN - 00.236
                                                    *TIME   - 16:08
                                                    *PREFIX - IMSDUMP
                                                    *TERMINAL- 3278
                                                    *PF KEYS - 24
                                                    *****

  B BPE       - IMS Connect BPE formatting

  X EXIT      - Exit IMS dump formatting

Enter END or RETURN command to terminate IMS Connect formatting.

```

Figure 98. IMS Connect Dump Formatting Menu Panel

Before you can format a dump, you need to initialize it. You initialize a dump as follows:

1. Select option 0 Show BPE status and initialize dump from the IMS Connect Dump Formatting menu panel. The IMS Connect initialization panel appears, as shown in Figure 99 on page 544, with the message that the symbol HWSCSD was not found and enter the IMS Connect jobname or ASID to cause IMS Connect IPCS symbols to be set for the dump.

```

----- IMS CONNECT DUMP CONTENT STATUS AND CONTROL -----
COMMAND ==>
SYMBOL HWSCSD NOT FOUND
Enter the IMS Connect jobname or ASID to cause the IMS
Connect IPCS symbols to be set for this dump. Press
ENTER with blank jobname and ASID fields to list all
IMS Connect address spaces in the dump.

          JOBNAME          ASID          A.S. TYPE          DUMPED?
-----
IMS Connect

SDWA Address:          BPE Release:
CSCD Address:          ??? Release:
??? Sys Name:          ??? Product:

```

Figure 99. IMS Connect Dump Initialization Panel

2. Specify either the jobname or the ASID of the IMS Connect address you want to format and press Enter. After you have provided a jobname or ASID, the remaining fields will be filled out in the initialization panel, as shown in Figure 100.

To obtain a list of *all* dumped IMS Connect address spaces, leave the jobname and ASID fields blank.

```

----- IMS CONNECT DUMP CONTENT STATUS AND CONTROL -----
COMMAND ==>

Enter the IMS Connect jobname or ASID to cause the IMS
Connect IPCS symbols to be set for this dump. Press
ENTER with blank jobname and ASID fields to list all
IMS Connect address spaces in the dump.

          JOBNAME          ASID          A.S. TYPE          DUMPED?
-----
IMS Connect  HWS71R          0042          HWS          YES

SDWA Address: 007C6920          BPE Release: 010300
CSCD Address: 06A01870          HWS Release: 010100
HWS Sys Name:          HWS Product: 5655-E51

```

Figure 100. IMS Connect Dump Initialization Panel after Initialization

3. After initializing the dump, Press F3 to return to the IMS Connect dump formatting primary menu.

After you have initialized a dump, you can then use the options on the IMS Connect dump formatting primary menu to browse the dump dataset, perform high- or low-level IMS Connect formatting, or perform IMS Connect BPE formatting.

---

## 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, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs

and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
J46A/G4  
555 Bailey Avenue  
San Jose, CA 95141-1003  
U.S.A.

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

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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 illustrates 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. 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 IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. \_enter the year or years\_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

---

## Trademarks

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

ACF/VTAM	IMS
BookManager	IMS/ESA
CICS	MVS
CICS/ESA	MVS/ESA
CICS/MVS	OS/390
DATABASE 2	RACF
DB2	RAMAC
DFS	RMF
DFS MS/MVS	VTAM
Hiperspace	z/OS
IBM	

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

---

## Product Names

In this book, the following licensed programs have shortened names:

- “DB2 for OS/390” is referred to as “DB2”.
- “COBOL for MVS & VS” is referred to as “COBOL”.
- “PL/I for MVS & VS” is referred to as “PL/I”.



## Bibliography

This bibliography lists all of the information in the IMS Version 9 library.

*3274 Control Unit Description and Programmer's Guide*, GA23-0061

*Advanced Communications Function for VTAM (ACF/VTAM) Macro Language Guide*, SC38-0256

*Advanced Communications Function for VTAM (ACF/VTAM) Macro Language Reference*, SC38-0261

*CICS Extended Recovery Facility Guide*, SC33-0661

*CICS Installation Guide*, SC33-0663

*CICS Operations Guide*, SC33-0668

*CICS Resource Definition (Macro)*, SC33-0667

*CICS System Definition Guide*, SC33-0664

*IBM DATABASE 2 Version 2 Administration Guide*, SC26-4374

*IBM DATABASE 2 Version 2 SQL Reference*, SC26-4380

*IMS/ESA CICS-IMS Database Control Guide*, SC33-1184

*IMS Online Recovery Service for z/OS User's Guide*, SC27-0944

*MVS/ESA Conversion Notebook for System Product Version 3, Vols. 1 and 2*, GC28-1567 and GC28-1568

*MVS/ESA Data Administration: Utilities*, SC26-4516

*DFSMS/MVS Program Management*, SC26-4916

*MVS/ESA Data Facility Product Version 3: Utilities*, GC26-4559

*MVS/ESA Interactive Problem Control System (IPCS) Planning and Customization*, GC28-1832

*MVS/ESA Interactive Problem Control System (IPCS) Users Guide*, GC28-1833

*MVS/ESA Job Control Language Reference*, GC28-1829

*MVS/ESA Job Control Language User's Guide*, GC28-1830

*MVS/ESA Linkage Editor and Loader User's Guide*, SC26-4510

*MVS/ESA Service Aids*, GC28-1844

*MVS/ESA System Programming Library: Application Development Guide*, GC28-1852

*OS PL/I Version 2: Programming Guide, Release 3*, SC26-4307

*Programming Language Considerations in an Information Management System (IMS) Environment*, G320-9538

*VS Cobol II Application Programming Guide*, SC26-4045

---

## IMS Version 9 Library

ZES1-2330	ADB	<i>IMS Version 9: Administration Guide: Database Manager</i>
ZES1-2331	AS	<i>IMS Version 9: Administration Guide: System</i>
ZES1-2332	ATM	<i>IMS Version 9: Administration Guide: Transaction Manager</i>
ZES1-2333	APDB	<i>IMS Version 9: Application Programming: Database Manager</i>
ZES1-2334	APDG	<i>IMS Version 9: Application Programming: Design Guide</i>
ZES1-2335	APCICS	<i>IMS Version 9: Application Programming: EXEC DLI Commands for CICS and IMS</i>
ZES1-2336	APTM	<i>IMS Version 9: Application Programming: Transaction Manager</i>
ZES1-2337	BPE	<i>IMS Version 9: Base Primitive Environment Guide and Reference</i>
ZES1-2338	CR	<i>IMS Version 9: Command Reference</i>
ZES1-2339	CQS	<i>IMS Version 9: Common Queue Server Guide and Reference</i>
ZES1-2340	CSL	<i>IMS Version 9: Common Service Layer Guide and Reference</i>
ZES1-2341	CG	<i>IMS Version 9: Customization Guide</i>
ZES1-2342	DBRC	<i>IMS Version 9: DBRC Guide and Reference</i>
ZES1-2343	DGR	<i>IMS Version 9: Diagnosis Guide and Reference</i>
ZES1-2344	FAST	<i>IMS Version 9: Failure Analysis Structure Tables (FAST) for Dump Analysis</i>
ZES1-2346	OLR	<i>IMS Version 9: HALDB Online Reorganization Guide</i>
ZES1-2380	CT	<i>IMS Version 9: IMS Connect Guide and Reference</i>
ZES1-2347	JGR	<i>IMS Version 9: IMS Java Guide and Reference</i>

ZES1-2348	IIV	<i>IMS Version 9: Installation Volume 1: Installation Verification</i>
ZES1-2349	ISDT	<i>IMS Version 9: Installation Volume 2: System Definition and Tailoring</i>
ZES1-2350	INTRO	<i>IMS Version 9: An Introduction to IMS</i>
ZES1-2351	MIG	<i>IMS Version 9: Master Index and Glossary</i>
ZES1-2352	MC1	<i>IMS Version 9: Messages and Codes, Volume 1</i>
ZES1-2353	MC2	<i>IMS Version 9: Messages and Codes, Volume 2</i>
ZES1-2354	OTMA	<i>IMS Version 9: Open Transaction Manager Access Guide and Reference</i>
ZES1-2355	OG	<i>IMS Version 9: Operations Guide</i>
GC17-7831	RPG	<i>IMS Version 9: Release Planning Guide</i>
ZES1-2358	URDBTM	<i>IMS Version 9: Utilities Reference: Database and Transaction Manager</i>
ZES1-2359	URS	<i>IMS Version 9: Utilities Reference: System</i>

**Supplementary Publications**

GC17-7825	LPS	<i>IMS Version 9: Licensed Program Specifications</i>
ZES1-2357	SOC	<i>IMS Version 9: Summary of Operator Commands</i>

**Publication Collections**

LK3T-7213	CD	IMS Version 9 Softcopy Library
LK3T-7144	CD	IMS Favorites
LBOF-7789	Hardcopy and CD	Licensed Bill of Forms (LBOF): IMS Version 9 Hardcopy and Softcopy Library
SBOF-7790	Hardcopy	Unlicensed Bill of Forms (SBOF): IMS Version 9 Unlicensed Hardcopy Library
SK2T-6700	CD	OS/390 Collection
SK3T-4270	CD	z/OS Software Products Collection
SK3T-4271	DVD	z/OS and Software Products DVD Collection

**Accessibility Titles Cited in this Book**

SA22-7787		z/OS V1R1.0 TSO Primer
SA22-7794		z/OS V1R1.0 TSO/E User's Guide
SC34-4822		z/OS V1R1.0 ISPF User's Guide, Volume 1



# Index

## Special characters

/DISPLAY POOL PSBP 5  
 /SSR command 478  
 /TRACE  
   ORS, tracing 403

## Numerics

2305 device 86, 119  
 2740 terminal  
   fast path terminal 143  
   LINE macro statement, specifying 111  
   LINEGRP macro statement, specifying 114  
   master terminal devices, choosing 23  
   system definition example 485, 492, 499  
   TERMINAL macro statement, specifying 143  
 2741 terminal  
   LINEGRP macro statement, specifying 114  
   system definition example 485, 493, 499  
   TERMINAL macro statement, specifying 143  
 2780 terminal  
   LINEGRP macro statement, specifying 114  
   TERMINAL macro statement, specifying 143  
 2848 control unit 79  
 2972 control unit 79  
 3270 terminal  
   LINE macro statement, specifying 111  
   LINEGRP macro statement, specifying 114  
   master terminal devices, choosing 23  
   TERMINAL macro statement, specifying 143  
 3271 control unit 79  
 3275 terminal  
   LINE macro statement, specifying 111  
   LINEGRP macro statement, specifying 114  
 3340 device 118  
 3350 device 118  
 3375 device 118  
 3380 device 118  
 3390 device 118  
 3600 work station 143

## A

ACBGGEN utility, PSB pools and 5, 69  
 ACCESS= parameter 81  
 ACTIV= parameter 436  
 ADDR= parameter 80, 112, 126, 140, 154  
 AGN= parameter 255, 449  
 AL= parameter 445  
 ALARM= parameter 358  
 ALIGN= parameter 140  
 allocating communication (MFS) pool space 7  
 ALOT= parameter 256, 454  
 ALTID= parameter 256  
 AOEXIT= parameter 72  
 AOI= parameter 177  
 AOI1= parameter 255

AOIP= parameter 256  
 AOIS= parameter 256  
 AOS= parameter 340  
 APARM= parameter 257  
 APF (authorized program facility)  
   specifying in batch procedures 194  
 APNDG= parameter 85  
 APPC= parameter 257  
 APPCASY= parameter 340  
 APPCIOT= parameter 340  
 APPCSE= parameter 257  
 APPLCTN macro  
   CCTL, use with 9  
   description 62  
   dynamic reassignment 63  
   MSNAME macro, and 66  
   ODBA, use with 9  
   parameters 64  
     FPATH= 64  
     GPSB= 64  
     LANG= 65  
     PGMTYPE= 65  
     PSB= 66  
     SCHDTYP= 66  
     SYSID= 66  
   syntax diagram 62  
   TRANSACT macro, and 62, 65, 66  
 APPLFE= parameter 257  
 application programs  
   online applications, defining 8  
   output limits 13  
   program resource requirements, defining 62  
 application programs, z/OS  
   using the ODBA (Open Database Access)  
     interface 465  
   using the Open Database Access (OBDA)  
     interface 465  
 APPLID= parameter 72  
 APPLID1= parameter 258  
 APPLID2= parameter 259  
 APPLID3= parameter 259  
 ARC= parameter 259, 421  
 ARCHDEF statement 407  
 ARCHDEF= parameter 369  
 ARCHIVE= parameter 371  
 ARMRST= parameter 259  
 ASM= parameter 102  
 ASMPRT= parameter 102  
 ASOT= parameter 259, 454, 459  
 ASSNCHANGE= parameter 340  
 authorized program facility (APF)  
   specifying in batch procedures 194  
 AUTLCHANGE= parameter 341  
 AUTLDESC= parameter 459  
 AUTLGN= parameter 459  
 AUTLID= parameter 459  
 AUTLMOD= parameter 459  
 AUTO= parameter 259, 358

**B**

BACKUP= parameter 80, 112, 116, 123, 126, 141, 155, 189, 455  
 batch dynamic allocation, disabling 409  
 batch initialization, modules that must be loaded 194  
 batch message processing  
   See see BMP  
 BFALLOC= parameter 83  
 BGWRT= parameter 391  
 BHBUF= parameter 439  
 binder order statements 50  
 BKO= parameter 259  
 BLDSNDX= parameter 391  
 BLOCK= parameter 371, 375, 376  
 BMP  
   declaring 11  
   dependent regions  
     useful dumps for 319  
   EXEC parameters 426  
 BMP dependent region, useful dumps for 319  
 BMP regions  
   choosing number 2  
 BMPUSID= parameter 341  
 BPECFG= parameter 260  
 BSIZ= parameter 260  
 BUF= parameter 260  
 buffer pool  
   defining OSAM subpools 393  
   defining VSAM subpools 387  
   description of 430  
   MFS 527  
   OSAM 430, 431  
   OSAM buffer pool compatibility 431  
   OSAM SB 430  
   sizes 416  
   specifying 430  
   VSAM 430  
 buffers  
   specifying  
     in DFSVSMxx member control statement 383  
 BUFFERS= parameter 119  
 BUFNO= parameter 408  
 BUFPOLS  
   description 2  
 BUFPOOLS macro  
   description 67  
   MFS considerations 523  
   parameters 67  
     DMB= 68  
     EPCB= 68  
     FORMAT= 68  
     FRE= 68  
   obsolete keywords 67  
     PSB= 69  
     PSBW= 69  
     SASPSB= 69  
   syntax diagram 67  
 BUFSETS= parameter 436  
 BUFSIZE= parameter 112, 126, 155

**C**

CBLTDLI procedure 313  
 CCTCVCAN= parameter 260  
 CCTL  
   DBCTL databases, and 446  
   preparing 447  
   starting 447  
 CFIRLM= parameter 410  
 CFNAMES= parameter 410  
 CFOSAM= parameter 410  
 CFVSAM= parameter 410  
 checkpoint  
   data sets 309  
   frequency of, setting 3  
 CHECKPOINT 3  
   restrictions 129  
 CICS  
   accessing IMS databases 471  
   data sharing 471  
   DBCTL, and 473  
   DL/I, and 473  
   intersystem communication, and 471  
   IRLM, and 471  
   PSB coding considerations 471  
   sequential buffering, and 472  
 CIOP= parameter 260  
 CKPTID= parameter 260  
 CL1=,CL2=CL3=,CL4= parameters 261  
 CLASS= parameter 261  
 CMDMCS= parameter 261  
 CMDSEC= parameter 335  
 CNBA= parameter 449  
 COBOL programs, coding 434  
 CODE= parameter 116, 134, 178  
 cold start  
   changing RECLNG 119  
   result of not specifying MODBLKS 87, 97  
 COMM macro  
   description 70  
   IMSGEN macro, and 136  
   MFS, and 523  
   parameters 71  
     APPLID= 72  
     COPYLOG= 72  
     EDTNAME= 73  
     FESEXIT= 73  
     MFSEXIT= 74  
     OPTIONS= 74  
     PASSWD= 76  
     RECANYS= 76  
     SECCNT= 78  
     SIMEXIT= 78  
   SECURITY macro, and 136  
   syntax diagram 70  
   TERMINAL macro, and 78  
   VTAM terminals, and 70, 72  
 command recognition character (CRC) 478  
 commands  
   /CHECKPOINT 3  
   /DISPLAY 5  
   /TRACE 482

- commands (*continued*)
    - DASD log activity, tracing 400
    - DB2 subsystem connection, tracing 405
    - DBF entries from FP, tracing 401
    - dispatcher activity, tracing 400
    - DL/I activity, tracing 400
    - Fast Path activity, tracing 401
    - latch activity, tracing 402
    - lock activity, tracing 402
    - options 399
    - ORS, tracing 403
    - OTMA control, tracing 403
    - queue manager, tracing 403
    - RSR log router, tracing 402
    - scheduler, tracing 404
    - shared queues interface, tracing 404
    - storage manager calls, tracing 404
    - CRC (command recognition character) 478
  - Common Queue Server
    - See see CQS
  - Common Service Layer (CSL)
    - DFSVSMxx
      - CSLT= 400
    - Member DFSCGxxx 334
    - Operations Manager (OM) 450
    - Resource Manager (RM) 450
    - Structured Call Interface (SCI) 450
  - communication devices
    - defining
      - intelligent Stations 18
      - non-VTAM terminals 14
      - switched devices 19
      - VTAM terminals 20
    - macros 19
      - CONFIG 19
      - IDLIST 19
      - LINE 19
      - LINEGRP 19
      - NAME 19
      - POOL 19
      - SUBPOOL 19
      - TERMINAL 19
  - COMPT= parameter 133, 155
  - COMPT1= parameter 156
  - COMPT2= parameter 156
  - COMPT3= parameter 156
  - COMPT4= parameter 156
  - COMPTn= parameter 455
  - CONFIG macro
    - description 78
    - IDLIST macro, and 78
    - Label field 79
    - LINE macro, and 78
    - LINEGRP macro, and 78
    - MFS 526
    - parameters 79
      - FEAT= 79
      - MODEL= 79
      - OPTIONS= 79
      - PTRSIZE= 79
    - syntax diagram 78
  - CONFIG= parameter 112
  - control intervals 277
  - control unit
    - 2848 79
    - 2972 79
    - 3271 79
    - CTLUNIT macro 79
  - CONVTYPE= parameter 415
  - COORD= parameter 477
  - COPYLOG= parameter 72
  - CORE= parameter 86, 262
  - coupling facility
    - enabling multiple DEDB areas to share same CF structure 384
    - OSAM data caching 411
    - structure names for Sysplex data sharing 409
  - CPLOG= parameter 86, 262
  - CPUTIME= parameter 262
  - CQS
    - address space 381
    - shared queues, placing messages on 296
  - CQS= parameter 381
  - CQSSSN= parameter 381
  - CRC (command recognition character) 88, 478
  - CRC= parameter 262, 478
  - CSAPSB= parameter 263
  - CSLG= parameter 264
  - CSLT= parameter 400
  - CTLUNIT macro
    - description 79
    - parameters 80
      - ADDR= 80
      - BACKUP= 80
      - MODEL= 80
    - syntax diagram 79
- ## D
- DASD
    - log tracing 400
    - logging 303, 304, 305, 407
  - data communication
    - EXEC parameters 424, 427
    - intelligent stations, defining 18
    - macros 490
    - master terminal, specifying 22
    - switched communication devices, defining 19
    - terminals, defining
      - non-VTAM terminals 14
      - VTAM terminals 13, 20, 21
  - data sharing
    - DATABASE macro statement 8
    - IMSCTRL macro statement 8
    - system configuration example 516
  - data transmission
    - transparent 155
  - DATABASE macro
    - CICS, and 473
    - description 80
    - online applications, defining 8
    - parameters 81

- DATABASE macro (*continued*)
  - ACCESS= 81
  - DBD= 82, 83
  - RESIDENT= 81
  - syntax diagram 81
- database recovery adapter (DRA)
  - setup 465
- database recovery adapter (DRA) start-up table
  - using with ODBA (Open Database Access) 465
  - using with Open Database Access (ODBA) 465
- database recovery control
  - See see DBRC
- Database Surveyor support 31
- databases
  - batch, tracking 195, 220
  - CICS, accessing with 471
  - DBRC, registering with 265
  - DL/I, using 202, 224
  - example 485
  - Fast Path 264, 271
  - main storage (MSDBs) 333
  - managing 80
  - online, declaring 9
- DB= parameter 437, 443, 445
- DB2
  - defining to IMS 475
  - IMS batch, and 483
  - modules 483
  - subsystem connection, tracing 405
  - subsystem identification parameters 420
  - tables, providing 483
- DBBBATCH procedure 195
  - DD statements 197
  - description 195
  - parameters 197
    - APARM= 257
    - BKO= 259
    - BUF= 260
    - DBRC= 265
    - EXCPVR= 271
    - FMTO= 272
    - GSGNAME= 274
    - HASH= 274
    - IMSID= 276
    - IMSPLEX= 277
    - IOB= 277
    - IRLM= 277
    - IRLMNM= 279
    - LOCKMAX= 281
    - LOGA= 281
    - LOGT= 281
    - MBR= 283
    - MON= 284
    - PRLD= 289
    - PSB= 289
    - RGN= 294
    - RGSUF= 294
    - RST= 294
    - SOUT= 297
    - SPIE= 297
    - SRCH= 298
- DBBBATCH procedure (*continued*)
  - parameters (*continued*)
    - SSM= 298
    - SWAP= 300
    - SYS= 300
    - SYS2= 300
    - TEST= 301
    - TMINAME= 301
- DBBBATCH procedures
  - parameters
    - CKPTID= 260
- DBBF= parameter 264
- DBC procedure
  - DD statements 203
  - description 198
  - parameters 202
    - AOIP= 256
    - ARC= 259
    - ARMRST= 259
    - AUTO= 259
    - BSIZ= 260
    - CCTCVCAN= 260
    - CIOP= 260
    - CMDMCS= 261
    - CORE= 262
    - CRC= 262
    - CSAPSB= 263
    - DBBF= 264
    - DBFP= 264
    - DBFX= 265
    - DBRCNM= 265
    - DBRSE= 266
    - DBWP= 266
    - DESC= 268
    - DLIDSIZE= 268
    - DLINM= 268
    - DLIPSB= 268
    - DMB= 269
    - DMHVF= 269
    - EPCB= 271
    - FDRMBR= 272
    - FIX= 272
    - FMTO= 272
    - FPDSSIZE= 272
    - FPWP= 274
    - IMSID= 276
    - IOB= 277
    - IRLM= 277
    - IRLMNM= 279
    - ISIS= 279
    - LGNR= 280
    - MAXPST= 283
    - MCS= 284
    - ORSMBR= 285
    - OTHR= 285
    - PIINCR= 288
    - PIMAX= 288
    - PRDR= 289
    - PREMSG= 289
    - PRLD= 289
    - PSBW= 290

- DBC procedure *(continued)*
  - parameters *(continued)*
    - PST= 290
    - RDMNM= 293
    - READNUM= 293
    - RES= 294
    - RGSUF= 294
    - RSRMBR= 294
    - SPM= 298
    - SRCH= 298
    - SSM= 298
    - SUF= 299
    - SVC2= 299
    - TRACK= 301
    - UHASH= 302
    - USERVAR= 302
    - VSPEC= 303
    - WADS= 303
    - WKAP= 303
    - YEAR4= 303
  - storing 193
- DBCTL
  - CCTL, and 446
  - CICS, and 473
  - environment, example 521
  - initializing 473
  - macros that apply 30
  - maximum number of BMP and DRA threads 3
  - using 30
- DBCTLID= parameter 449
- DBD= parameter 82, 264, 334, 394
- DBFMSDBx
  - description 333
  - Main Storage Databases, specifying 333
  - parameters 333
- DBFP= parameter 264
- DBFX= parameter 265
- DBLDL= parameter 265
- DBRC
  - DATABASE macro, ACCESS parameter 81
  - procedure, description 203
  - procedure, storing 193
- DBRC procedure
  - automatic initiation 204
  - manual initiation 204
  - parameters 205
    - DD statements 205
    - DPRTY= 269
    - IMSID= 276
    - IMSPLEX= 277
    - RGN= 294
    - SOUT= 297
    - SYS2= 300
  - PDS member name 204
    - DBRCNM= 204
  - START commands
    - example 205
- DBRC= parameter 89, 265
- DBRCNM= parameter 89, 265, 421
- DBRSE= parameter 266
- DBWP= parameter 266
- DC macro
  - examples 490
  - macros 490
- DC= parameter 266
- DCC parameters
  - parameters
    - ARMRST= 259
- DCC procedure
  - DD statements 214
  - description 205
  - parameters 213
    - ALOT= 256
    - AOIP= 256
    - AOIS= 256
    - APPC= 257
    - APPCSE= 257
    - APPLID1= 258
    - APPLID2= 259
    - APPLID3= 259
    - ARC= 259
    - ARMRST= 259
    - ASOT= 259
    - AUTO= 259
    - CHTS= 260
    - CIOP= 260
    - CMDMCS= 261
    - CRC= 262
    - DC= 266
    - DESC= 268
    - DLQT= 269
    - DPRTY= 269
    - DSCT= 270
    - EMHB= 270
    - EMHL= 270
    - EPCB= 271
    - ETO= 271
    - EXVR= 272
    - FBP= 272
    - FESTIM= 272
    - FIX= 272
    - FMTO= 272
    - FRE= 274
    - GRNAME= 274
    - GRSNAME= 274
    - HIOP= 275
    - HSBID= 275
    - HSBMBR= 276
    - IMSID= 276
    - ISIS= 279
    - LGMSGSZ= 280
    - LHTS= 281
    - LOGT= 281
    - LTERM= 282
    - LUMC= 282
    - LUMP= 282
    - MAXPST= 283
    - MCS= 284
    - NHTS= 284
    - NLXB= 284
    - ORSMBR= 285
    - OTMA= 285

DCC procedure (*continued*)  
 parameters (*continued*)

OTMANM= 286  
 OTMASP= 286  
 PASSWD= 287  
 PASSWD1= 287  
 PRDR= 289  
 PRLD= 289  
 PSB= 289  
 PSBW= 290  
 PST= 290  
 QBUF= 291  
 QBUFMAX= 291  
 QBUFSZ= 292  
 QTL= 292  
 QTU= 292  
 RCF= 293  
 RECA= 293  
 RECASZ= 293  
 RES= 294  
 RGN= 294  
 RGSUF= 294  
 RSRMBR= 294  
 RVFY= 294  
 SAV= 294  
 SGN= 295  
 SHAREDQ= 296  
 SHMSGSZ= 297  
 SOD= 297  
 SOUT= 297  
 SPM= 298  
 SRCH= 298  
 SSM= 298  
 SUF= 299  
 SVC2= 299  
 SYS= 300  
 SYS1= 300  
 SYS2= 300  
 TRACK= 301  
 TRN= 301  
 TSR= 302  
 UHTS= 302  
 USERVAR= 302  
 VAUT= 302  
 VSPEC= 303  
 WADS= 303  
 WKAP= 303  
 YEAR4= 303

storing 193

DCCTL

macros that apply 31  
 using 31

DCLWA= parameter 90, 178  
 DD= parameter 437  
 DDNAME= parameter 116, 127, 406, 449  
 DEDBMAS= parameter 385  
 DEFERFIX= parameter 358  
 defining external subsystems to IMS 475  
 defining language interface module 481  
 DEGRADE= parameter 408

dependent regions

BMP  
 useful dumps for 319  
 IFP (IMS Fast Path)  
 useful dumps for 319  
 IMS Fast Path (IFP)  
 useful dumps for 319

DESC= parameter 90, 268

design limit 60

devices

2305 86  
 3340 86  
 3350 86  
 3375 86  
 3380 86  
 3390 86  
 LGDK 86

DFS62DTx

description 414  
 parameters 415

DFSAERA0 module 466

DFSAERG0 module 466

DFSAERM0 module 466

DFSCDLI0 module 466

DFSDCxxx

data communications options, specifying 338  
 description 338  
 parameters 338  
 validating with Syntax Checker 529

DFSDFRnn

description 350  
 parameters 350

DFSDFRxx

description 351  
 parameters 351

DFSFIXnn

description 352  
 parameters 352

DFSHALDB

single partition processing 463

DFSHSBxx

description 357, 361  
 parameters 357, 361  
 preinitialization routines, specifying dependent region 361  
 XRF options 357

DFSIDEF macro 420

DFSIDEF0 macro 420

DFSINTxx

description 361  
 parameters 361

DFSJBP procedure

DD statements 216

description 215

parameters 216

ENVIRON= 270

JVMPOPMAS= 279

DFSJMP procedure

DD statements 218

description 217

parameters 218

- DFSJMP procedure (*continued*)
  - ENVIRON= 270
  - JVMOPWKR= 280
- DFSLI macro 481
- DFSLI000 (language interface module) 481
- DFSMPLxx
  - description 363
  - high-use program modules, making resident 363
  - parameters 364
- DFSMPR procedure
  - DD statements 220
  - description 219
  - IMSMSG job 249
  - parameters 219
    - AGN= 255
    - ALTID= 256
    - APARM= 257
    - APPLFE= 257
    - CL1=,CL2=CL3=,CL4= 261
    - DBLDL= 265
    - IMSID= 276
    - LOCKMAX= 281
    - NBA= 284
    - OBA= 284
    - OPT= 285
    - OVLA= 287
    - PCB= 287
    - PREINIT= 289
    - PRLD= 289
    - PWFI= 291
    - RGN= 294
    - SOD= 297
    - SOUT= 297
    - SPIE= 297
    - SSM= 298
    - STIMER=, message-driven programs 298
    - STIMER=, non-message driven programs 299
    - SYS2= 300
    - TLIM= 301
    - VALCK= 302
    - VFREE= 303
    - VSFX= 303
- DFSORSxx member 365
- DFSPBDBC
  - description 366
  - parameters 366
- DFSPBDCC
  - description 366
  - parameters 366
- DFSPBIMS
  - description 366
  - parameters 366
- DFSPBxxx
  - validating with Syntax Checker 529
- DFSRESLIB DD statement
  - IMS procedures, specifying in 194
- DFSRSRxx
  - description 366
  - parameters 367
- DFSSPMxx
  - description 377
- DFSSPMxx (*continued*)
  - examples 380
  - parameters 378
- DFSSQxxx
  - description 381
  - parameters 381
  - validating with Syntax Checker 529
- DFSSTAT report 305
- DFSVMxxx
  - control statements, types of 383
  - description 383
  - Fast Path DEDB buffer pools, defining 384
  - Fast Path parallel database open, disabling 409
  - OSAM buffer pools, defining 392
  - OSAM subpools, defining 392
  - sequential buffering, specifying 395
  - serviceability and trace options, defining 396
  - VSAM buffer pools, defining 385
  - VSAM performance options 391
  - VSAM subpools, defining 387
- DIRCA= parameter 268
- DISP= parameter 400
- DISPLAY command 5
- DL/I
  - accounting procedures 6
  - call image trace 405
  - CICS, and 473
  - COBOL interface 313
  - DBC procedure, and 202
  - DLISAS procedure, modifying 6
  - execution parameters, system 419
  - exits, modifying 6
  - interfaces 313
  - invoking 6
  - lock activity, tracing 402
  - PL/I interface 313
  - security considerations 7
  - selecting 4
  - storage considerations 4
  - tuning considerations 7
- DL/I= parameter 400
- DLIBATCH procedure
  - DD statements 222
  - description 220
  - parameters 222
    - APARM= 257
    - BKO= 259
    - BUF= 260
    - CKPTID= 260
    - DBRC= 265
    - EXCPVR= 271
    - FMTO= 272
    - GSGNAME= 274
    - HASH= 274
    - IMSID= 276
    - IMSPLEX= 277
    - IOB= 277
    - IRLM= 277
    - IRLMN= 279
    - LOCKMAX= 281
    - LOGA= 281

DLIBATCH procedure *(continued)*parameters *(continued)*

LOGT= 281  
 MBR= 283  
 MON= 284  
 PRLD= 289  
 RGN= 294  
 RGSUF= 294  
 RST= 294  
 SOUT= 297  
 SPIE= 297  
 SRCH= 298  
 SSM= 298  
 SWAP= 300  
 SYS2= 300  
 TEST= 301  
 TMINAME= 301

DLIDSIZE= parameter 268, 371

DLINM= parameter 91, 268

DLIPSB= parameter 268

## DLISAS procedure

automatic initiation 223

DD statements 224

description 224

IMS procedure, and 231

manual initiation 223

parameters 224

DPRTY= 269

IMSID= 276

RGN= 294

SOUT= 297

SYS2= 300

PDS member name 223

DLINM= 223

RACF protected resources 223

resource protection 223

storing 193

DLOG= parameter 400

DLQT= parameter 269

DMB= parameter 68, 269

DMHVF= parameter 269

DOPT online change 63

DPM (distributed presentation management)

MFS SLU P terminal 155

DPRTY= parameter 269

DRA (database recovery adapter)

setup 465

DRA (database recovery adapter) start-up table

using with ODBA (Open Database Access) 465

using with Open Database Access (OBDA) 465

DREF storage, defining 350

DSCT= parameter 270

DSETS= parameter 119

DSN= parameter 374, 375

DSNAME= parameter 449

DSSIZE= parameter 372

dump formatter 541

DUMP= parameter 400

## dumping

batch system

SYSABEND 319

dumping *(continued)*batch system *(continued)*

SYSMDUMP to a data set 319

SYSUDUMP 319

dependent region 319

installing 315

IRLM, and 321

online 316

SDUMP to a SYS1.DUMPxx data set 316

Spinoff dump to SYSABEND 317

Spinoff dump to SYSUDUMP 317

SYSABEND 316

SYSMDUMP 317

SYSUDUMP 316

## options

available in an IMS environment 315

in effect in an IMS environment 316

system execution parameters 419

z/OS options 316

SDUMP 316

SYSABEND 316

SYSMDUMP, batch system only 316

SYSMDUMP, online system only 316

SYSUDUMP 316

DUMPIO= parameter 401

## dumps

BMP regions 319

FMTO options 317

FMTO options, batch 320

IFP regions 319

MPP regions 319

## DXRJPROC procedure

DD statements for IRLM 226

parameters

DEADLOK= 266

parameters for IRLM 226

startup procedure for IRLM 225

## dynamic allocation

terminals 25

## dynamic PSB 11

dynamic reassignment and APPLCTN macro 63

**E**

EDIT= parameter 116, 133, 159, 179, 189, 455

EDTNAME= parameter 73

EMHB= parameter 270

EMHL= parameter 270

EMHQ= parameter 381

ENVIRON= parameter 270

EPCB= parameter 68, 271

ESMT (external subsystem module table)

defining 477

ESMT= parameter 477

## ETO

## descriptors

creating, rules for 451

format 451

logon 452

MFS device 456

MSC 457



ETO (*continued*)descriptors (*continued*)

overrides 452

user 457

IMSCTRL macro, ETOFEAT 91, 450

including in IMS 25, 450

stage 1 50

terminal naming rules 60

ETO= parameter 271

ETOFEAT= parameter 91

EVAL= parameter 440

examples

coding conventions 62

Data Communication macro statements 490

data sharing system configuration 516

IMS DB/DC system definition 485

IMS DBCTL environment 521

MSC 511

exclusive access to databases 81

EXCPVR= parameter 271

execution parameters

categories and purpose 426

data communication 417, 424, 427

database 416

db buffer sizes 416

db performance options 417

dc buffer sizes 417

dc performance options 418

performance-related 425

PSB-related 427

recovery-related 420, 426, 429

region control 424, 428

security-related 422, 426, 429

specifying 415

system

active regions 418

DL/I address space 419

IRLM options 419

nucleus identifier 418

performance options 418

z/OS options 418

Extended Terminal Option

See see ETO

external subsystem attach facility

adding an SSM PROCLIB member 475

CRC 478

defining external subsystems to IMS 475

defining language interface module 481

DFSLI macro 481

DFSLI000 477

IMS.PROCLIB, description 475

LIT 477, 481

OPTIONS statement 482

REO 477

RTT 477

specifying desired trace options 482

SSM= EXEC parameter 479

SUBS= parameter 482

EXVR= parameter 272

**F**

F= parameter 334

Fast Database Recovery

See see FDBR

Fast Path

DBF trace entries 401

DEDB buffers pools, defining 384

IMSFP procedure 248

macro statements

APPLCTN 64, 65

DATABASE 80

FPCTRL 82

parallel database open, disabling 409

terminals

2740 terminal 143

FAST= parameter 401

FBP= parameter 272

FDBR

enabling a DB/DC subsystem for 26

enabling a DBCTL subsystem for 26

FDR procedure 226

FDRMBR parameter 26

FDR procedure

DD statements 230

parameters 229

ARC= 259

ARMRST= 259, 260

BSIZ= 260

CSAPSB= 263

CSLG= 264

DBBF= 264

DBWP= 266

DESC= 268

DFRMBR= 272

DLIPSB= 268

DMB= 269

DPRTY= 269

FMTO= 272

IMSID= 276

IMSPLEX= 277

IRLMNM= 279

LGNR= 280

MCS= 284

PSB= 289

PSBW= 290

RGN= 294

RGSUF= 294

SOUT= 297

SPM= 298

SUF= 299

SVC2= 299

SYS1= 300

SYS2= 300

UHASH= 302

VSPEC= 303

WADS= 303

WKAP= 303

FDRMBR= parameter 272

FEAT= parameter 79, 116, 141, 159, 455

FES exit routine 73

FESEXIT= parameter 73

FESTIM= parameter 272  
 fetch request elements  
     See see FRE  
 Field Edit exit routine 74  
 Finance work station 144  
 FIX= parameter 272  
 FIXBLOCK= parameter 387  
 FIXDATA= parameter 386  
 fixed pages, defining 352  
 FIXINDEX= parameter 386  
 FMTO= parameter 272  
 FORMAT= parameter 68  
 forms control  
     vertical 156  
 FPATH= parameter 64, 179  
 FPBOF= parameter 449  
 FPBUF= parameter 161, 449  
 FPCTRL  
     description 2, 82  
     parameters 83  
         BFALLOC= 83  
         INREXIT= 83  
         OTHREAD= 84  
         USRHASH= 84  
     syntax diagram 82  
 FPDSSIZE= parameter 272, 372  
 FPOPN= parameter 272  
 FPRLM= parameter 273  
 FPTT= parameter 401  
 FPUTIL procedure  
     description 230  
     example 230  
     parameters 231  
         AGN= 255  
         ALTID= 256  
         DBD= 264  
         DIRCA= 268  
         IMSID= 276  
         PRLD= 289  
         REST= 294  
         RGN= 294  
         SOUT= 297  
         SSM= 298  
         SYS2 300  
 FPWP= parameter 274  
 FRE  
     controlling 7  
     estimating 7  
     MFS, and 527  
     parameter, BUFPOOLS macro 68  
 FRE= parameter 68, 274  
 FUNCLV= parameter 449

## G

GPSB= parameter 64  
 GRAFFIN= parameter 341  
 GRESTAE= parameter 342  
 GRNAME= parameter 274  
 GRSNAME= parameter 274  
 GSGNAME= parameter 91, 274, 372

## H

HALDB  
     single partition processing 463  
 HALDB partition  
     cold starting IMS 82  
     redefining in ALL or MODBLKS system definition 82  
     system definition types  
         ALL 41  
         MODBLKS 41  
 hash table slots, specifying 462  
 HASH= parameter 274  
 HEADCTL= parameter 161  
 HIDAM databases  
     declaring 9  
     macro instructions 80  
 High Availability Large Database (HALDB) partition  
     cold starting IMS 82  
     redefining in ALL or MODBLKS system definition 82  
     system definition types  
         ALL 41  
         MODBLKS 41  
 high-speed sequential control statements, specifying  
     examples 445  
     SETO 442  
     SETR 444  
     syntax 442  
 high-use program modules, making resident with  
     DFSMPLEX 363  
 HIOP= parameter 275  
 HSB= parameter 92  
 HSBID= parameter 275  
 HSBMBR= parameter 276  
 HSn= parameter 389

## I

IBM 3480 tape drive  
     using with IMS log data sets 245  
 IC= parameter 443, 445  
 ICOMPT= parameter 133  
 IDC0= parameter 401  
 IDLE command 129  
 IDLIST macro  
     CONFIG macro, and 78  
     description 84  
     label 84  
     parameters 84  
         configname 84  
         id 84  
     syntax diagram 84  
 IDS= parameter 113  
 IFP (IMS Fast Path)  
     dependent regions  
         useful dumps for 319  
 IFP (IMS Fast Path) dependent region, useful dumps  
     for 319  
 ILTMODE= parameter 372  
 IMS  
     Offline Dump Formatter 541

IMS Connect  
   dump formatter 541  
 IMS Fast Path (IFP)  
   dependent regions  
     useful dumps for 319  
 IMS Fast Path (IFP) dependent region, useful dumps  
   for 319  
 IMS procedure  
   and the DLISAS procedure 231  
   DD statements 241  
   description 232  
   parameters 240  
     ALOT= 256  
     AOI1= 255  
     AOIS= 256  
     APPC= 257  
     APPCSE= 257  
     APPLID1= 258  
     APPLID2= 259  
     APPLID3= 259  
     ARC= 259  
     ARMRST= 259  
     ASOT= 259  
     AUTO= 259  
     BSIZ= 260  
     CCTCVCAN= 260  
     CIOP= 260  
     CMDMCS= 261  
     CPLOG= 262  
     CRC= 262  
     CSAPSB= 263  
     DBBF= 264  
     DBFP= 264  
     DBFX= 265  
     DBRCNM= 265  
     DBWP= 266  
     DC= 266  
     DESC= 268  
     DLIDSIZE= 268  
     DLINM= 268  
     DLIPSB= 268  
     DLQT= 269  
     DMB= 269  
     DMHVF= 269  
     DPRTY= 269  
     DSCT= 270  
     EMHB= 270  
     EMHL= 270  
     EPCB= 271  
     ETO= 271  
     EXVR= 272  
     FBP= 272  
     FDRMBR= 272  
     FESTIM= 272  
     FIX= 272  
     FMTO= 272  
     FPDSSIZE= 272  
     FPOPEN= 272  
     FPRLM= 273  
     FPWP= 274  
     FRE= 274  
   IMS procedure (continued)  
   parameters (continued)  
     GRNAME= 274  
     GRSNAME= 274  
     HIOP= 275  
     HSBID= 275  
     HSBMBR= 276  
     IMSGROUP= 276  
     IMSID= 276  
     IOVFI= 277  
     IRLM= 277  
     IRLMNM= 279  
     ISIS= 279  
     LGMSGSZ= 280  
     LGNR= 280  
     LHTS= 281  
     LOGT= 281  
     LSO= 281  
     LTERM= 282  
     LUMC= 282  
     LUMP= 282  
     MCS= 284  
     MNPS= 284  
     MNPSPW= 284  
     PRDR= 289  
     PRLD= 289  
     PSB= 289  
     PSBW= 290  
     PST= 290  
     QBUF= 291  
     QBUFMAX= 291  
     QBUFSZ= 292  
     QTL= 292  
     QTU= 292  
     RCF= 293  
     RCFTCB= 293  
     RDMNM= 293  
     READNUM= 293  
     RECA= 293  
     RECA SZ= 293  
     RES= 294  
     RGN= 294  
     RGSUF= 294  
     RSRMBR= 294  
     RVFY= 294  
     SAV= 294  
     SGN= 295  
     SHAREDQ= 296  
     SHMSGSZ= 297  
     SOD= 297  
     SOUT= 297  
     SPM= 298  
     SRCH= 298  
     SSM= 298  
     SUF= 299  
     SVC2= 299  
     SVSODR= 299  
     SYS= 300  
     SYS1= 300  
     SYS2= 300  
     TCORACF= 300

- IMS procedure *(continued)*
  - parameters *(continued)*
    - TRACK= 301
    - TRN= 301
    - TSR= 302
    - UHASH= 302
    - UHTS= 302
    - USERVAR= 302
    - VAUT= 302
    - VSPEC= 303
    - WADS= 303
    - WKAP= 303
    - YEAR4= 303
  - PSB pools 232
  - RACF protected resources 231
  - sample 232
  - storing 193
  - z/OS, and 433
- IMS PROCLIB members
  - DFSORSxx 365
- IMS system
  - designing 1
- IMS transactions
  - defining 12
  - TRANSACTION macro, using 12
- IMS.LGMSG data set 118
- IMS.PROCLIB members
  - altering 329
  - DBFMSDBx 333
  - DFS62DTx 414
  - DFSDCxxx 338
  - DFSDFRxx 351
  - DFSFIXnn 352
  - DFSHSBxx 357
  - DFSINTxx 361
  - DFSPBDBC 366
  - DFSPBDCC 366
  - DFSPBIMS 366
  - DFSRSRxx 366
  - DFSSPMxx 377
  - DFSSQxxx 381
  - DFSVSMxx 383
  - execution, preparing for 333
  - generated 327
  - initializing 330
  - modifications, controlling 332
  - renaming 329
  - specifying 333
  - tailoring 327, 330
- IMS.QBLKS data set 118
- IMS.SHMSG data set 118
- IMSBATCH procedure 195, 219, 220, 230, 242, 244, 248, 250, 254
  - DD statements 243
  - parameters 243
    - AGN= 255
    - ALTID= 256
    - APARM= 257
    - CKPTID= 260
    - CPUTIME= 262
    - DIRCA= 268
- IMSBATCH procedure *(continued)*
  - parameters *(continued)*
    - IMSID= 276
    - IMSPLEX= 277
    - IN= 277
    - LOCKMAX= 281
    - MBR= 283
    - NBA= 284
    - OBA= 284
    - OPT= 285
    - PAGES= 287
    - PARDLI= 287
    - PREINIT= 289
    - PRLD= 289
    - PSB= 289
    - RGN= 294
    - SOUT= 297
    - SPIE= 297
    - SSM= 298
    - STIMER=, message-driven programs 298
    - STIMER=, non-message driven programs 299
    - SYS2= 300
    - TEST= 301
- IMSCOBGO procedure
  - and the IBM 3480 tape drive 245
  - DD statements 246
  - description 244
  - parameters 246
    - BKO= 259
    - BUF= 260
    - CKPTID= 260
    - DBRC= 265
    - EXCPVR= 271
    - FMTO= 272
    - IMSID= 276
    - IMSPLEX= 277
    - IOB= 277
    - IRLM= 277
    - IRLMNM= 279
    - LOGA= 281
    - LOGT= 281
    - MBR= 283
    - MON= 284
    - PRLD= 289
    - PSB= 289
    - RGN= 294
    - RST= 294
    - SOUT= 297
    - SPIE= 297
    - SRCH= 298
    - SWAP= 300
    - SYS2= 300
    - TEST= 301
- IMSCOBOL procedure
  - DD statements 247
  - description 246
  - parameters 247
    - MBR= 283
    - SOUT= 297
    - SYS2= 300

- IMSCTF
  - description 2
  - setting checkpoints 3
- IMSCTF macro
  - description 85
  - parameters 85
    - APNDG= 85
    - CORE= 86
    - CPLOG= 86
    - LOG= 86
    - PRDR= 86
    - RDS= 86
    - SVCNO= 86
  - syntax diagram 85
- IMSCTRL
  - description 2
  - selecting a lock manager 4
  - use without specifying MODBLKS 87
- IMSCTRL macro
  - defining BMP regions 2
  - defining message regions 2
  - defining regions
    - BMP 2
      - message 2
  - description 87
  - parameters 88
    - CMDCHAR= 88
    - DBRC= 89, 420
    - DBRCNM= 89
    - DCLWA= 90
    - DESC= 90
    - DLINM= 91
    - ETOFEAT= 91
    - GSGNAME= 91
    - HSB= 92
    - IMSID= 92
    - IRLM= 92
    - IRLMNM= 93
    - MAXCLAS= 93
    - MAXIO= 93
    - MAXREGN= 93
    - MCS= 94
    - MODBLKS= 96
    - MSVERIFY 97
    - MSVID= 94
    - NAMECHK= 94
    - NUCLEUS 97
    - ON-LINE 97
    - RSRFEAT= 95
    - SYSTEM= 95
  - syntax diagram 87
  - system definition types 29
- IMSFP procedure
  - DD statements 248
  - description 248
  - parameters 248
    - AGN= 255
    - ALOT= 256
    - ALTID= 256
    - APARM= 257
    - CPUTIME= 262
- IMSFP procedure (*continued*)
  - parameters (*continued*)
    - DBLDL= 265
    - DIRCA= 268
    - IMSID= 276
    - LOCKMAX= 281
    - MBR= 283
    - NBA= 284
    - OPT= 285
    - PREINIT= 289
    - PRLD= 289
    - PSB= 289
    - RGN= 294
    - SOD= 297
    - SOUT= 297
    - SSM= 298
    - STIMER=, message-driven programs 298
    - STIMER=, non-message driven programs 299
    - SYS2= 300
    - TLIM= 301
- IMSGEN
  - description 2
- IMSGEN macro
  - assembler and binder options 102
  - COMM macro, and 136
  - communication options parameters
    - MFSTEST= 109
    - SYSMSG= 109
  - data set option parameters
    - MACLIB= 104
    - MACSYS= 104
    - MODGEN= 104
    - NODE= 104
    - OBJDSET= 105
    - PROCLIB= 105
    - SCEERUN= 106
    - UMAC0= 106
    - UMACx= 106
    - USERLIB= 106
  - description 98
  - general communication options 109
  - IMS data set options 104
  - JCL statement parameters
    - JCL= 107
    - JOBCTL= 106
    - JPRTY= 107
    - MFSDFMT= 107
    - ONEJOB= 107
    - SCL= 108
    - UJCLx= 108
  - JCL statements 106
  - MFS, and 523
  - parameters 101
    - ASM= 102
    - ASMPRT= 102
    - LKPRT= 102
    - LKRGN= 102
    - LKSIZE= 103
    - SUFFIX= 103
    - TERM= 103
    - UPDTPRT= 104

- IMSGEN macro (*continued*)
    - sample IMSGEN macro statement 110
    - SECURITY macro, and 136
    - security options 109
    - security options parameters
      - PSWDSEC= 109
      - SECCNT= 109
      - TERMSEC= 109
    - syntax diagram 98
  - IMSGROUP= parameter 276
  - IMSID= parameter 92, 276
  - IMSMSG job 249
  - IMSplex
    - managing 450
  - IMSPLEX macro
    - data set option parameters
      - IMSPLEX= 104
  - IMSPLEX= parameter 104, 277, 336
  - IMSPLI procedure
    - DD statements 250
    - parameters 250
      - MBR= 283
      - SPIE= 297
      - SYS2= 300
  - IMSPLIGO procedure
    - DD statements 252
    - description 250
    - parameters 252
      - BKO= 259
      - BUF= 260
      - CKPTID= 260
      - DBRC= 265
      - EXCPVR= 271
      - FMTO= 272
      - IMSID= 276
      - IMSPLEX= 277
      - IOB= 277
      - IRLM= 277
      - IRLMNM= 279
      - LOGA= 281
      - LOGT= 281
      - MGR= 283
      - MON= 284
      - PRLD= 289
      - PSB= 289
      - RGN= 294
      - RST= 294
      - SOUT= 297
      - SPIE= 297
      - SRCH= 298
      - SWAP= 300
      - SYS2= 300
      - TEST= 301
  - IMSRDR procedure
    - DD statement 253
    - description 252
    - parameters 253
      - CLASS= 261
      - MBR= 283
      - SYS2= 300
    - storing 193
  - IMSRDR procedure (*continued*)
    - z/OS, and 433
  - IMSWT= parameter 342
  - IN= parameter 277
  - INQ= parameter 180
  - INQUIRY= parameter 134, 180
  - INREXIT= parameter 83
  - INSERT= parameter 392
  - intelligent stations
    - defining 18
  - Interactive Dump Formatter
    - DD concatenations to update 315
    - dump options
      - available 315
      - in effect 316
    - installing 315
  - IOB= parameter 277
  - IOSCB= parameter 395
  - IOVF control intervals 277
  - IOVFI= parameter 277
  - IRLM
    - CICS, and 471
    - coupling facility structure names 409
    - DEADLOK parameter 266
    - dumping 321
    - execution parameters 419
    - IMSCTRL macro
      - IRLM parameter 92
      - IRLMNM parameter 93
      - LOCKTIME parameter 413
      - performance and DEADLOK parameter 267
      - SDUMP 321
      - startup procedure 225
    - IRLM= parameter 92, 277
    - IRLMGRP= parameter 278
    - IRLMID= parameter 278
    - IRLMN= parameter 279
    - IRLMNM= parameter 93
    - ISC surveillance link 164, 172
    - ISIS= parameter 279
    - ISSUE681= parameter 401
    - ISSUE840= parameter 402
- ## J
- JCL= parameter 107
  - JCLIN 52
  - job accounting 107
  - JOBCTL= parameter 106, 107
  - JVMOPMAS=
    - parameter description 279
  - JVMOPWKR=
    - parameter description 280
  - JVMOPWKR= parameter 280
  - JVMPOPAS= parameter 279
- ## K
- KEYEVENT= parameter 358

**L**

- label 84
- label field 79
- LANG= parameter 65
- language interface module 481
- Large System generation
  - See see LGEN
- LATC= parameter 402
- LBUFMAX= parameter 372
- LGDK device 86
- LGEN
  - configuration 30, 45
  - preprocessor 43
  - SMP/E JCLIN 52
  - stage 1 processing 42
  - stage 2 processing 43
  - storage requirements 47, 48
  - system definition 42, 48
- LGMSGSZ= parameter 280
- LGNR= parameter 280
- LHTS= parameter 281
- LIBATCH procedure
  - parameters
    - PSB= 289
- LINE macro
  - CONFIG macro, and 78
  - description 111
  - parameters 111
    - ADDR= 112
    - BACKUP= 112
    - BUFSIZE= 112
    - CONFIG= 112
    - IDS= 113
    - MODEL= 113
    - RESP= 113
    - VERIFY= 113
  - syntax diagram 111
  - SYSIN Local Card Reader 114
- LINEGRP macro
  - CONFIG macro, and 78
  - description 114
  - parameters 115
    - BACKUP= 116
    - CODE= 116
    - DDNAME= 116
    - EDIT= 116
    - FEAT= 116
    - MODE= 117
    - UNITYPE= 117
  - syntax 114
- LIT (Language Interface Token) 481
- LIT= parameter 476, 481
- LKPRT= parameter 102
- LKRGN= parameter 102
- LKSIZE= parameter 103
- LMODE= parameter 372
- LNK= parameter 358
- lock manager, selecting
  - IRLM 4
    - program isolation 4
- lock timeout function, IRLM 413
- LOCK= parameter 402
- LOCKMAX= parameter 281
- LOCKSEC= parameter 342
- LOCKTAB= parameter 281
- LOG= parameter 86, 359, 406
- LOGA= parameter 281
- LOGCOUNT= parameter 373
- logical term name 129, 131
- logical terminal
  - See see LTERM
- LOGT= parameter 281
- LRTT= parameter 402
- LSO (local storage option)
  - DL/I address space, specifying 4
  - PSB pools, defining 5
  - storage considerations 4
- LSO= parameter 281
- LTE= parameter 281
- LTERM
  - POOL macro 134
    - specifying 129
- LTERM= parameter 161, 282
- LU 6 terminal
  - TERMINAL macro statement, specifying 143
  - VTAM devices 141
- LUMC= parameter 282
- LUMI= parameter 402
- LUMP= parameter 282
- LUNAME= parameter 415

**M**

- MACLIB= parameter 104
- macro statements
  - APPLCTN
    - description 62
    - online programs, declaring 9
- BUFPOOLS
  - description 67
  - MFS pool size, controlling 7
- coding conventions 62
- COMM 70
- CONFIG 78
- CTLUNIT 79
- DATABASE
  - data sharing 8
  - online databases, declaring 9
- FPCTRL 82
  - guidelines 59
- IDLIST 84
- IMSCTF 85
- IMSCTRL
  - checkpoint frequency, setting 3
  - data sharing 8
  - description 87
  - LGEN system definition 42
- IMSGEN 98
- LINE 111
- LINEGRP 114
- maximum occurrences 60
- MSGQUEUE 118

- macro statements (*continued*)
  - MSLINK 123
  - MSNAME 124
  - MSPLINK 125
  - NAME 129
  - POOL 134
  - resource naming rules 60
  - RTCODE 134
  - SECURITY 135
  - SPAREA 140
  - STATION 140
  - SUBPOOL 141
  - system configuration macros
    - BUFPOOLS 2
    - FPCTRL 2
    - IMSCTF 2
    - IMSCTRL 2
    - IMSGEN 2
    - MSGQUEUE 2
    - SECURITY 2
  - system configuration macros, use of 2
  - system definition macros, use of 1, 55
  - TERMINAL
    - 2740 terminal 143
    - 2741 terminal 143
    - 2780 terminal 143
    - 3270 terminal 143
    - 3600 work station 143
    - Finance work station 143
    - LU 6 143
    - SLU 1 143
    - SLU 2 143
    - SLU P 143
    - SPOOL 143
    - System/3 terminal 143
    - System/7 terminal 143
  - TRANSACT
    - description 174
    - IMS transactions, defining 12
  - TYPE 188
  - types of 1
  - VTAMPOOL 190
- macro table, using 31
- MACSYS= parameter 104
- Main Storage Database
  - See see MSDB
- master terminal
  - configuring 23
  - devices, choosing 22, 23
  - secondary logging 24
  - secondary, commands that can be copied to 73
  - specifying 22, 129, 132
- MAXCLAS= parameter 93
- MAXCSA= parameter 283
- MAXIO= parameter 93
- MAXPST= parameter 283
- MAXREGN= parameter 93
- MAXRGN= parameter 180
- MAXSB= parameter 396
- MAXTHRD= parameter 449
- MAXUSRS= parameter 283
- MBR= parameter 283
- MCS= parameter 94, 284
- message format pool
  - size, estimating 7
  - size, specifying 417
- Message Format Service (MFS)
  - pool space
    - BUFPOOLS macro 7
- Message Formatting Service
  - See see MFS
- message prefix
  - sizes
    - Version 7 120
    - Version 8 120
    - Version 9 120
- Message Processing Region
  - data communication EXEC parameters 424, 427
  - parameters, categories and purposes 422
  - performance-related parameters 425
  - PSB-related EXEC parameters 423, 427
  - recovery-related parameters 420, 426, 429
  - region-control EXEC parameters 424, 428
  - security-related EXEC parameters 422, 426, 429
- message queue
  - data set allocation 118
  - data sets
    - IMS.LGMSG 118
    - IMS.QBLKS 118
    - IMS.SHMSG 118
- message regions
  - choosing number 2
- MFS
  - buffer pool use 527
  - BUFPOOLS macro 523
  - COMM macro 523
  - CONFIG macro 526
  - FREs 527
  - IMSGEN macro 523
  - INDEX directory 526
  - MSGQUEUE macro 523
  - pool space, allocating 7
  - pool space, defining 7
  - system definition and programming
    - considerations 523
  - TERMINAL 524
  - TYPE macro 524
- MFSDFMT= parameter 107
- MFSEXIT= parameter 74
- MFSTEST (message format service test facility) 75, 109
- MFSTEST= parameter 109
- MILEINTV= parameter 372
- MINTHRD= parameter 449
- MNPS= parameter 284, 359
- MNPSPW= parameter 284, 360
- MODBLKS
  - use IMSCTRL without specifying 87, 97
- MODBLKS= parameter 96
- MODE= parameter 117, 180, 371, 375, 376, 408, 415
- MODEL= parameter 79, 80, 113, 162
- MODETBL= parameter 123, 127, 162



MODGEN= parameter 104  
 MODSTAT= parameter 377  
 MODSTAT2= parameter 377  
 modules  
   DFSAERA0 466  
   DFSAERG0 466  
   DFSAERM0 466  
   DFSCDLI0 466  
 MON= parameter 284  
 MRQPSBN= parameter 119  
 MSC  
   APPLCTN macro, and 63  
   example 511  
   macro statements, preparing 19  
   macros  
     MSLINK 20  
     MSNAME 20  
     MSPLINK 20  
     NAME 20  
   VTAM buffer size, restrictions 127  
 MSCSEC= parameter 342  
 MSCT= parameter 403  
 MSDB  
   DBFMSDBx member 333  
 MSDB= parameter 284  
 MSDBABND= parameter 333  
 MSGDEL= parameter 142, 163  
 MSGQ= parameter 382  
 MSGQUEUE macro  
   description 2, 118  
   MFS, and 523  
   parameters 119  
     BUFFERS= 119  
     DSET= 119  
     MRQPSBN= 119  
     RECLNG= 120  
     SHUTDWN= 122  
   syntax diagram 118  
 MSGTYPE= parameter 181  
 MSLINK macro  
   description 123  
   parameters 123  
     BACKUP= 123  
     MODETBL= 123  
     MSPLINK= 124  
     OPTIONS= 124  
     PARTNER= 124  
   syntax diagram 123  
 MSNAME macro  
   description 124  
   label field 125  
   parameters 125  
     SYSID= 125  
   syntax diagram 124  
 MSPLINK macro  
   description 125  
   parameters 126  
     ADDR= 126  
     BACKUP= 126  
     BUFSIZE= 126  
     DDNAME= 127

MSPLINK macro (*continued*)  
   parameters (*continued*)  
     MODETBL= 127  
     NAME= 128  
     OPTIONS= 128  
     SESSION= 128  
     TYPE= 129  
   syntax diagram 125  
 MSPLINK= parameter 124  
 MSVERIFY= parameter 97  
 MSVID= parameter 94  
 MTOID= parameter 373  
 multi-area structures  
   defining DEDB buffer pools 384  
 Multiple Systems Coupling  
   See see MSC

## N

NAME macro  
   description 129  
   parameters 131  
     COMPT= 133  
     EDIT= 133  
     ICOMPT= 133  
   lterm 131  
     OUTPUT= 133  
     remote lterm name 132  
   syntax diagram 130  
 NAME= parameter 128, 141, 142, 164  
 NAMECHK= parameter 94  
 NBA= parameter 284  
 NBRSEGS= parameter 334  
 NHTS= parameter 284  
 NLXB= parameter 284  
 NODE= parameter 104  
 NOIC= parameter 443  
 nonswitched communication lines  
   described by LINE macro 111  
 NOPROCH= parameter 443  
 NTO device  
   TERMINAL macro statement 143  
 NUC= parameter 356  
 nucleus definition 96  
 NUCLEUS= parameter 97

## O

OBA= parameter 284  
 OBDA (Open Database Access)  
   AGN security 467  
   application programs, binding 466  
   APSB security 469  
   ICMD security 469  
   RAS security 468  
   security  
     AGN 467  
     APSB 469  
     ICMD 469  
     RAS 468  
 OBJDSET= parameter 105

OCMD= parameter 403  
 ODBA (Open Database Access)  
   accessing IMS databases 465  
   defining PSB names 9  
   defining security 466  
   interface 465  
   modules  
     which ones to place in JOBLIB 466  
     which ones to place in STEPLIB 466  
   overview 465  
   setup 465  
 ODBASE= parameter 285  
 OLC= parameter 337  
 OLCSTAT= parameter 337  
 OLDS= parameter 408  
 OMPROC= parameter 338  
 ON-LINE= parameter 97  
 ONEJOB= parameter 107  
 online databases, declaring 9  
 online programs  
   characteristics 10  
   declaring 9  
 Open Database Access (OBDA)  
   AGN security 467  
   application programs, binding 466  
   APSB security 469  
   ICMD security 469  
   modules  
     which ones to place in JOBLIB 466  
     which ones to place in STEPLIB 466  
   RAS security 468  
   security  
     AGN 467  
     APSB 469  
     ICMD 469  
     RAS 468  
 Open Database Access (ODBA)  
   accessing IMS databases 465  
   defining PSB names 9  
   defining security 466  
   interface 465  
   overview 465  
   setup 465  
 Operations Manager (OM) 450  
 OPT= parameter 285  
 OPTIONS statement 482  
 OPTIONS= parameter 74, 79, 124, 128, 164, 189,  
   455, 460  
 order statements, binder 50  
 ORS (Online Recovery System)  
   activating trace for 403  
   tracing activity 403  
 ORS= parameter 285  
 ORSMBR= parameter 285  
 ORTT= parameter 403  
 OSAM  
   buffer pools, defining 392, 430, 431  
   coupling facility 410, 411  
   I/O errors, tracing 401  
   sequential buffering 433  
   subpools, defining 392, 432

OSAMGTF= parameter 395  
 OTHR= parameter 285  
 OTHREAD= parameter 84  
 OTMA  
   control flow, tracing 403  
   DCC procedure 205  
   generic resource (GRNAME) parameter 274  
   message prefix sizes 120  
 OTMA= parameter 285  
 OTMANM= parameter 286  
 OTMASE= parameter 286  
 OTMASP= parameter 286  
 OTMT= parameter 403  
 OUT= parameter 287  
 OUTBUF= parameter 169, 456  
 OUTPUT= parameter 133  
 OVLA= parameter 287

## P

PAGES= parameter 287  
 parallel session support 190  
 parameters  
   ACCESS= 81  
   ACTIV= 436  
   ADDR= 80, 112, 126, 140, 154  
   AGN= 255, 449  
   AL= 445  
   ALARM= 358  
   ALIGN= 140  
   ALOT= 256, 454  
   ALTID= 256  
   AOEXIT= 72  
   AOI= 177  
   AOI1= 255  
   AOIP= 256  
   AOIS= 256  
   AOS= 340  
   APARM= 257  
   APNDG= 85  
   APPC= 257  
   APPCASY= 340  
   APPCLOT= 340  
   APPCSE= 257  
   APPLFE= 257  
   APPLID= 72  
   APPLID1= 258  
   APPLID2= 259  
   APPLID3= 259  
   ARC= 259, 421  
   ARCHDEF= 369  
   ARCHIVE= 371  
   ARMRST= 259  
   ASM= 102  
   ASMPRT= 102  
   ASOT= 259, 454, 459  
   ASSNCHANGE= 340  
   AUTLCHANGE= 341  
   AUTLDESC= 459  
   AUTLGN= 459  
   AUTLID= 459

## parameters (continued)

AUTLMOD= 459  
 AUTO= 259, 358  
 BACKUP= 80, 112, 116, 123, 126, 141, 155, 189, 455  
 BFALOC= 83  
 BGWRT= 391  
 BHBUF= 439  
 BKO= 259  
 BLDSNDX= 391  
 BLOCK= 371, 375, 376  
 BMPUSID= 341  
 BPECFG= 260  
 BSIZ= 260  
 BUF= 260  
 BUFFERS= 119  
 BUFNO= 408  
 BUFSETS= 436  
 BUFSIZE= 112, 126, 155  
 CCTVCAN= 260  
 CFIRLM= 410  
 CFNAMES= 410  
 CFOSAM= 410  
 CFVSAM= 410  
 CIOP= 260  
 CKPTID= 260  
 CL1=,CL2=CL3=,CL4= 261  
 CLASS= 261  
 CMDMCS= 261  
 CMDSEC= 335  
 CNBA= 449  
 CODE= 116, 134, 178  
 COMPT= 133, 155  
 COMPT1= 156  
 COMPT2= 156  
 COMPT3= 156  
 COMPT4= 156  
 COMPTn= 455  
 CONFIG= 112  
 CONVTYPE= 415  
 COORD= 477  
 COPYLOG= 72  
 CORE= 86, 262  
 CPLOG= 86, 262  
 CPUTIME= 262  
 CQS= 381  
 CQSSN= 381  
 CRC= 262, 478  
 CSAPSB= 263  
 CSLG= 264  
 CSLT= 400  
 DB= 437, 443, 445  
 DBBF= 264  
 DBCTLID= 449  
 DBD= 82, 264, 334, 394  
 DBFP= 264  
 DBFX= 265  
 DBLDL= 265  
 DBRC= 89, 265  
 DBRCNM= 89, 265, 421  
 DBRSE= 266

## parameters (continued)

DBWP= 266  
 DC= 266  
 DCLWA= 90, 178  
 DD= 437  
 DDNAME= 116, 127, 406, 449  
 DEDBMAS= 385  
 DEFERFIX= 358  
 DEGRADE= 408  
 DESC= 90, 268  
 DIRCA= 268  
 DISP= 400  
 DL/I= 400  
 DLIDSIZE= 268, 371  
 DLINM= 91, 268  
 DLIPSB= 268  
 DLOG= 400  
 DLQT= 269  
 DMB= 68, 269  
 DMHVF= 269  
 DPRTY= 269  
 DSCT= 270  
 DSETS= 119  
 DSN= 374, 375  
 DSNAME= 449  
 DSSIZE= 372  
 DUMP= 400  
 DUMPIO= 401  
 EDIT= 116, 133, 159, 179, 189, 455  
 EDTNAME= 73  
 EMHB= 270  
 EMHL= 270  
 EMHQ= 381  
 ENVIRON= 270  
 EPCB= 68, 271  
 ESMT= 477  
 ETO= 271  
 ETOFEAT= 91  
 EVAL= 440  
 EXCPVR= 271  
 EXVR= 272  
 F= 334  
 FAST= 401  
 FBP= 272  
 FDRMBR= 272  
 FEAT= 79, 116, 141, 159, 455  
 FESEXIT= 73  
 FESTIM= 272  
 FIX= 272  
 FIXBLOCK= 387  
 FIXDATA= 386  
 FIXINDEX= 386  
 FMTO= 272  
 FORMAT= 68  
 FPATH= 64, 179  
 FPBOF= 449  
 FPBUF= 161, 449  
 FPDSSIZE= 272, 372  
 FPOPN= 272  
 FPRLM= 273  
 FPTT= 401

## parameters (continued)

FPHP= 274  
 FRE= 68, 274  
 FUNCLV= 449  
 GPSB= 64  
 GRAFFIN= 341  
 GREATAE= 342  
 GRNAME= 274  
 GRSNAME= 274  
 GSGNAME= 91, 274, 372  
 HASH= 274  
 HEADCTL= 161  
 HIOP= 275  
 HSB= 92  
 HSBID= 275  
 HSBMBR= 276  
 HSn= 389  
 IC= 443, 445  
 ICOMPT= 133  
 IDC0= 401  
 IDS= 113  
 ILTMODE= 372  
 IMSGROUP= 276  
 IMSID= 92, 276  
 IMSPLEX= 104, 277, 336  
 IMSWT= 342  
 IN= 277  
 INQ= 180  
 INQUIRY= 134, 180  
 INREXIT= 83  
 INSERT= 392  
 IOB= 277  
 IOSCB= 395  
 IOVFI= 277  
 IRLM= 92, 277  
 IRLMGRP= 278  
 IRLMID= 278  
 IRLMN= 279  
 IRLMNM= 93  
 ISIS= 279  
 ISSUE681= 401  
 ISSUE840= 402  
 JCL= 107  
 JOBCTL= 106, 107  
 JVMOPWKR= 280  
 JVMOPMAS= 279  
 KEYEVENT= 358  
 LANG= 65  
 LATC= 402  
 LBUFMAX= 372  
 LGMSGSZ= 280  
 LGNR= 280  
 LHTS= 281  
 LIT= 476, 481  
 LKPRT= 102  
 LKRGN= 102  
 LKSIZE= 103  
 LMODE= 372  
 LNK= 358  
 LOCK= 402  
 LOCKMAX= 281

## parameters (continued)

LOCKSEC= 342  
 LOCKTAB= 281  
 LOG= 86, 359, 406  
 LOGA= 281  
 LOGCOUNT= 373  
 LOGT= 281  
 LRTT= 402  
 LSO= 281  
 LTE= 281  
 LTERM= 161, 282  
 LUMC= 282  
 LUMI= 402  
 LUMP= 282  
 LUNAME= 415  
 MACLIB= 104  
 MACSYS= 104  
 MAXCLAS= 93  
 MAXCSA= 283  
 MAXIO= 93  
 MAXPST= 283  
 MAXREGN= 93  
 MAXRGN= 180  
 MAXSB= 396  
 MAXTHRD= 449  
 MAXUSRS= 283  
 MBR= 283  
 MCS= 94, 284  
 MFSDFMT= 107  
 MFSEXIT= 74  
 MFSTEST= 109  
 MILEINTV= 372  
 MINTHRD= 449  
 MNPS= 284, 359  
 MNPPSW= 284, 360  
 MODBLKS= 96  
 MODE= 117, 180, 371, 375, 376, 408, 415  
 MODEL= 79, 80, 113, 162  
 MODETBL= 123, 127, 162  
 MODGEN= 104  
 MODSTAT= 377  
 MODSTAT2= 377  
 MON= 284  
 MRQPSBN= 119  
 MSCSEC= 342  
 MSCT= 403  
 MSDB= 284  
 MSDBABND= 333  
 MSGDEL= 142, 163  
 MSGQ= 382  
 MSGTYPE= 181  
 MSPLINK= 124  
 MSVERIFY= 97  
 MSVID= 94  
 MTOID= 373  
 NAME= 128, 141, 142, 164  
 NAMECHK= 94  
 NBA= 284  
 NBRSEGS= 334  
 NHTS= 284  
 NLXB= 284

## parameters (continued)

no longer supported  
 AOEXIT= 72  
 NODE= 104  
 NOIC= 443  
 NOPROCH= 443  
 NUC= 356  
 NUCLEUS= 97  
 OBA= 284  
 OBJDSET= 105  
 obsolete  
 PASSWD 287  
 OCMD= 403  
 ODBASE= 285  
 OLC= 337  
 OLCSTAT= 337  
 OLDS= 408  
 OMPROC= 338  
 ON-LINE= 97  
 ONEJOB= 107  
 OPT= 285  
 OPTIONS= 74, 79, 124, 128, 164, 189, 455, 460  
 ORS= 285  
 ORSMBR= 285  
 ORTT= 403  
 OSAMGTF= 395  
 OTHR= 285  
 OTHREAD= 84  
 OTMA= 285  
 OTMANM= 286  
 OTMASE= 286  
 OTMASP= 286  
 OTMT= 403  
 OUT= 287  
 OUTBUF= 169, 456  
 OUTPUT= 133  
 OVLA= 287  
 PAGES= 287  
 PARDLI= 287  
 PARLIM= 182  
 PARTNER= 124  
 PASSWD= 76, 137, 287  
 PASSWD1= 287  
 PCB= 287, 437, 443, 445  
 PGMTYPE= 65  
 PGPROT= 288  
 PIINCR= 288  
 PIMAX= 288  
 PMTO= 343  
 PMTO1-8= 343  
 PMTOG= 344  
 POOLID= 386  
 PRDR= 86, 289  
 PREINIT= 289  
 PREMSG= 289  
 PRICOUNT= 370, 374, 376  
 PRIUNIT= 370, 374, 375  
 PRLD= 289  
 PROCLIB= 105  
 PROCLIM= 183  
 PRTY= 184

## parameters (continued)

PSB= 66, 69, 289, 437  
 PSBW= 69, 290  
 PSDEPAB= 360  
 PST= 290  
 PSTIMER= 344  
 PSWDSEC= 109  
 PTRSIZE= 79, 171, 456  
 PU= 171  
 PWFI= 291  
 QBUF= 291  
 QBUFHITH= 291, 292  
 QBUFLWTH= 291  
 QBUFMAX= 291  
 QBUFSZ= 292  
 QMGR= 403  
 QTL= 292, 421  
 QTU= 292, 421  
 RCF= 293  
 RCFTCB= 293  
 RCLASS= 137, 344  
 RCVYCONV= 344, 460  
 RCVYFP= 345, 461  
 RCVYSTSN= 345, 461  
 RDMNM= 293  
 RDS= 86, 360  
 READNUM= 293  
 RECA= 293  
 RECANY= 76  
 RECASZ= 293  
 RECLNG= 120  
 REDO= 372  
 REO= 477  
 RES= 294  
 RESIDENT= 81  
 RESP= 113  
 REST= 294  
 RETPD= 373  
 RGN= 294  
 RGSUF= 294  
 RLDSDEF= 374  
 RMENV= 337  
 RNR= 346  
 ROUTING= 185  
 RRS= 294  
 RRSST= 403  
 RSENAME= 360  
 RSR= 375  
 RSRFEAT= 95  
 RSRMBR= 294  
 RST= 294  
 RTT= 477  
 RVFY= 294  
 SAPPLID= 346  
 SASPSB= 69  
 SAV= 294  
 SBBUF= 440  
 SBBUFCB= 439  
 SBONLINE= 396  
 SCEERUN= 106  
 SCHD= 185, 404

## parameters (continued)

SCHDTYP= 66  
 SCIPROC= 338  
 SCL= 108  
 SCOPE= 295  
 SECCNT= 78, 109, 137  
 SECCOUNT= 370, 374, 376  
 SECLVL= 137  
 SECUNIT= 370, 374, 376  
 SEGNO= 186  
 SEGSIZE= 172, 186, 456  
 SERIAL= 186  
 SESSION= 128, 172  
 SGN= 295  
 SHAREDQ= 296  
 SHMSGSZ= 297  
 SHUTDOWN= 122  
 SIDE= 415  
 SIGNON= 346  
 SIMEXIT= 78  
 SIZE= 172  
 SLDSDEF= 375  
 SLU2= 347  
 SMTO= 347  
 SMTO1-8= 347  
 SMTOG= 347  
 SOD= 297, 449  
 SOUT= 297  
 SPA= 187  
 SPACE= 370, 375, 376  
 SPAP= 297  
 SPIE= 297  
 SPM= 298  
 SQGROUP= 382  
 SQTT= 404  
 SRCH= 298  
 SRMDEF= 347, 461  
 SSM= 298  
 SSN= 476  
 SST= 476  
 START= 406, 440  
 STIMER=  
     message-driven programs 298  
     non-message driven programs 299  
 STM= 348  
 STOP= 406, 440  
 STRG= 404  
 STRINGMX= 392  
 STRINGNM= 387  
 SUBS= 405  
 SUF= 299  
 SUFFIX= 103  
 SURV= 360  
 SVC2= 299  
 SVCNO= 86  
 SVSODR= 299  
 SWAP= 300  
 SWITCH= 361  
 SYNCLEVEL= 415  
 SYS= 300  
 SYS1= 300

## parameters (continued)

SYS2= 300  
 SYSID= 66, 125, 187  
 SYSMMSG= 109  
 SYSTEM= 95  
 TCORACF= 300  
 TERM= 103  
 TERMNL= 138  
 TERMSEC= 109  
 TEST= 301  
 TIME= 373  
 TIMEOUT= 449  
 TIMER= 450  
 TLIM= 301  
 TMINAME= 98, 301, 376  
 TPNAME= 415  
 TRACE= 301  
 TRACK= 301  
 TRANCMD= 138  
 TRANSLM= 141  
 TRKMODS= 377  
 TRN= 301  
 TRUNC= 348  
 TSR= 302  
 TYPE= 129, 131, 139, 141, 174, 481  
 UHASH= 302  
 UHTS= 302  
 UJCL1= 108  
 UJCL2= 108  
 UJCL3= 108  
 UJCL4= 108  
 UJCL5= 108  
 UJCLx= 108  
 UMAC0= 106  
 UMAC1= 106  
 UMAC2= 106  
 UMAC3= 106  
 UNIT= 174, 370, 374, 375, 456  
 UNITYPE= 117, 190, 456  
 UPDTPRT= 104  
 USERID= 450  
 USERLIB= 106  
 USERVAR= 302, 361  
 USRHASH= 84  
 VACBOPN= 349  
 VALCK= 302  
 VAUT= 302  
 VERIFY= 113  
 VFREE= 303  
 VSAMFIX= 392  
 VSAMPLS= 392  
 VSFx= 303  
 VSPEC= 303  
 WADS= 303, 408, 421  
 WADSDEF= 408  
 WFI= 188  
 WKAP= 303  
 XPLINK= 303  
 YEAR4= 303  
 PARDLI= parameter 287  
 PARLIM= parameter 182

- PARTNER= parameter 124
- passtickets, using instead of passwords 346
- PASSWD= parameter 76, 137, 287
- PASSWD1= parameter 287
- PCB= parameter 287, 437, 443, 445
- performance-related EXEC parameters 425
- PGMTYPE= parameter 65
- PGPROT= parameter 288
- PIINCR= parameter 288
- PIMAX= parameter 288
- PL/I modules, organizing 433
- PLITDLI procedure
  - description 313
- PMTO= parameter 343
- PMTO1-8= parameter 343
- PMTOG= parameter 344
- polling features 117
- POOL macro 134
- POOLID= parameter 386
- PRDR= parameter 86, 289
- PREINIT= parameter 289
- preinitialization routines, specifying in DFSHSBxx 361
- PREMSG= parameter 289
- preprocessor
  - executing 45
  - JCL 45
  - LGEN output 45
  - parameters, defining 47
  - resource names
    - verified 44
  - storage requirements, estimating 47
- PRICOUNT= parameter 370, 374, 376
- PRIUNIT= parameter 370, 374, 375
- PRLD= parameter 289
- procedures
  - CBLTDLI 313
  - DBBBATCH 195
  - DBC 198
  - DBRC 203
  - DCC 205
  - DD statements 303
    - DFSCTL 304
    - DFSDB2AF 304
    - DFSDUPPM 304
    - DFSESL 304
    - DFSHALDB 304
    - DFSOLPnn 304
    - DFSOLSnn 304
    - DFSRESLB 304
    - DFSSTAT 305
    - DFSTCF 305
    - DFSVSAMP 305
    - DFSWADSn 305
    - FORMATA 306
    - FORMATB 306
    - FPTRACE 306
    - IEFRDER 306
    - IEFRDER2 306
    - IMS 306
    - IMSACB 306
    - IMSACBA 306
  - procedures (*continued*)
    - DD statements (*continued*)
      - IMSACBB 306
      - IMSIRD 306
      - IMSLOGR 307
      - IMSMON 307
      - IMSRDS 307
      - IMSRDS2 307
      - IMSTFMTA 307
      - IMSTFMTB 307
      - JVAERR 307
      - JVAIN 307
      - JVAOUT 307
      - JCLOUT 307
      - JCLPDS 307
      - LGMSG 307
      - LGMSG1 308
      - MATRIXA 308
      - MATRIXB 308
      - MODBLKSA 308
      - MODBLKSB 308
      - MODSTAT 308
      - MODSTAT2 309
      - MSDBCP1 309
      - MSDBCP2 309
      - MSDBCP3 309
      - MSDBCP4 309
      - MSDBDUMP 309
      - MSDBINIT 309
      - OLCSTAT 309
      - PRINTDD 309
      - PROCLIB 310
      - QBLKS 310
      - QBLKSL 310
      - RECONn 310
      - SHMSG 310
      - SHMSG1 310
      - STEPLIB 310
      - SYSABEND 310
      - SYSHALDB 310
      - SYSIN 310
      - SYSLIB 311
      - SYSLIN 311
      - SYSLMOD 310
      - SYSPRINT 310
      - SYSTSPRT 310
      - SYSUDUMP 310
      - SYSUTn 311
    - DFSJBP 215
    - DFSJMP 217
    - DFSMPR 219
    - DLIBATCH 220
    - DLISAS 224
    - FDR 226
    - FPUTIL 230
    - IMS 232
    - IMSBATCH 242
    - IMSCOBGO 244
    - IMSCOBOL 246
    - IMSFP 248
    - IMSPLI 249

procedures (*continued*)  
 IMSPLIGO 250  
 IMSRDR 252  
 parameters 255  
 PLITDLI 313  
 RDIBATCH 254  
 storing 193  
 sysgen-supplied, cataloged 191, 193  
 PROCLIB  
 See see IMS.PROCLIB members  
 PROCLIB= parameter 105  
 PROCLIM= parameter 183  
 program modules, making high-use resident 363  
 programs  
 application, characteristics of 10  
 online, declaring 9  
 PRTY= parameter 184  
 PSB  
 calculating EPCB storage 68  
 EPCB storage  
 calculating 68  
 pools  
 defining 5  
 PSB performance options  
 choosing 10  
 dynamic 11  
 not resident 10  
 resident 10  
 PSB pools, defining  
 /DISPLAY POOL PSBP command 5  
 ACBGEN utility 5  
 BUFPOOLS macro 67  
 performance options 10  
 PSB-related EXEC parameters 423, 427  
 PSB= parameter 66, 69, 289, 437  
 PSBW= parameter 69, 290  
 PSDEPAB= parameter 360  
 PST= parameter 290  
 PSTIMER= parameter 344  
 PSWDSEC= parameter 109  
 PTRSIZE= parameter 79, 171, 456  
 PU= parameter 171  
 PWF= parameter 291

## Q

QBUF= parameter 291  
 QBUFHITH= parameter 291, 292  
 QBUFLWTH= parameter 291  
 QBUFMAX= parameter 291  
 QBUFSZ= parameter 292  
 QMGR= parameter 403  
 QTL= parameter 292, 421  
 QTU= parameter 292, 421

## R

RACF  
 DLISAS procedure 223  
 passtickets, using 346  
 RCF= parameter 293

RCFTCB= parameter 293  
 RCLASS= parameter 137, 344  
 RCVYCONV= parameter 344, 460  
 RCVYFP= parameter 345, 461  
 RCVYSTSN= parameter 345, 461  
 RDIBATCH procedure  
 DD statements 255  
 description 253  
 parameters 254  
 APARM= 257  
 BKO= 259  
 BUF= 260  
 CKPTID= 260  
 DBRC= 265  
 EXCPVR= 271  
 FMTO= 272  
 IMSID= 276  
 IMSPLEX= 277  
 IOB= 277  
 IRLM= 277  
 IRLMNM= 279  
 LOCKMAX= 281  
 LOGA= 281  
 MBR= 283  
 MON= 284  
 PRLD= 289  
 PSB= 289  
 RGN= 294  
 RST= 294  
 SOUT= 297  
 SPIE= 297  
 SRCH= 298  
 SSM= 298  
 SWAP= 300  
 SYS= 300  
 SYS2= 300  
 TEST= 301  
 RDMNM= parameter 293  
 RDS= parameter 86, 360  
 read access to databases 82  
 read-only access to databases 82  
 READNUM= parameter 293  
 RECA= parameter 293  
 RECANY= parameter 76  
 RECASZ= parameter 293  
 RECLNG= parameter 120  
 recommendations  
 checkpoint frequency  
 take frequent checkpoints to minimize restart  
 time 3  
 databases  
 register with DBRC not DFSMDA 240  
 deadlock frequency  
 choose value high enough to resolve deadlocks  
 and low enough not to impact performance 267  
 choose value low enough not to impact  
 performance 267  
 deadlocks  
 monitor number of deadlocks and adjust local  
 deadlock value accordingly 267



- recommendations (*continued*)
  - exit routines
    - lower-numbered exit routines should be field exit routines 74
    - lower-numbered exit routines should be segment exit routines 74
  - Fast Path (FPCTRL macro)
    - specify BLOCKS=FP to avoid performance problems and system abends 353
  - if running XRF
    - code HSB=YES on the IMSCTRL macro 92
  - IMS batch
    - do not place DFSVSAMP in the same PDS as your user application files 383
  - IMSGEN macro
    - do not specify the assembler option COMPAT(CASE) 102
    - use NOUSING and FLAG(NOPUSH) options to avoid assembler errors. 102
  - IRLM startup procedures
    - specify same value for DEADLOK parameters to avoid last IRLM value becoming the value for all 267
  - JAVAERR DD statements
    - specify to take advantage of function 307
  - JAVAOUT DD statements
    - specify to take advantage of function 307
  - language interface module
    - CSECT name should match the bind name 481
  - LINEGRP macro
    - allocate at least two data sets 116
  - MSC and shared queues are used and IMSs are different release levels
    - use highest calculated values for size1 and size2 for each IMS. 121
  - RSR
    - If RSR is not installed or enabled, do not specify a value for GSGNAME 91
  - secondary master logging
    - specify ALL, unless other terminals besides the master are issuing many commands 24
  - security
    - migrate your current AGN SMU-based security implementation to a RAS security implementation 139
    - protect IMS data sets from unauthorized access by using RACF or an equivalent product 529
    - use OM (Operations Manager) command security rather than IMS security 336
  - serviceability
    - specify ON for the DL/I=, LOCK=, DISP=, and SCHD=parameters 396
  - system log data set (SLDS)
    - specify PRICOUNT(0) and SECCOUNT(0) to limit to 5 volumes or less during restart 370, 374, 376
  - tracing
    - run the FPTRACE in a test environment only 401
  - transactions
    - use upper-case characters for TPNAMES 415
  - recovery service
    - PROCLIB member (DFSORSxx) 365
  - recovery-related EXEC parameters 420, 426, 429
  - REDO= parameter 372
  - region-control EXEC parameters 424, 428
  - regions, dependent
    - dumping options 319
    - number of, choosing the 2, 93, 290
    - preinitialization routines 361
  - Remote Site Recovery
    - See see RSR
  - REO 477
  - REO= parameter 477
  - reports
    - HALDB single partition processing 463
  - RES= parameter 294
  - RESIDENT online change 63
  - RESIDENT= parameter 81
  - Resource Management Facility
    - See see RMF
  - Resource Name table, storage 48
  - resource naming
    - rules
      - ETO terminals 60
      - macros 60
      - node names 60
      - subpool names 60
  - resource naming rules
    - ETO terminals 60
    - macros 60
  - RESP= parameter 113
  - REST= parameter 294
  - restarting IMS
    - checkpoints, using 3
  - RETPD= parameter 373
  - RGN= parameter 294
  - RGSUF= parameter 294
  - RLDSDEF= parameter 374
  - RMENV= parameter 337
  - RMF
    - accounting procedures 6
  - RNR= parameter 346
  - ROUTING= parameter 185
  - RRS= parameter 294
  - RRST= parameter 403
  - RSENAME= parameter 360
  - RSR
    - COMM macro
      - APPLID parameter 72
      - PSWD parameter 76
    - DFSRSRxx member 366
    - DL/I batch jobs, 195
    - enabling 91
    - IMSCTRL macro, RSRFEAT parameter
      - GSGNAME parameter 91
      - RSRFEAT parameter 95
      - TMINAME parameter 98
    - including in IMS 26
    - log router activity, tracing 402
    - options, specifying 366
  - RSR= parameter 375

RSRFEAT= parameter 95  
 RSRMBR= parameter 294  
 RST= parameter 294  
 RTCODE macro  
   description 134  
   parameters 134  
     CODE= 134  
     INQUIRY= 134  
   syntax diagram 134  
 RTT (resource translation table) 477  
 RTT= parameter 477  
 RVFY= parameter 294

## S

SAPPLID= parameter 346  
 SASPSB= parameter 69  
 SAV= parameter 294  
 SBBUF= parameter 440  
 SBBUFCB= parameter 439  
 SBCO 438  
 SBESNAP 440  
 SBIC 438  
 SBONLINE control statement 395  
 SBONLINE= parameter 396  
 SBPARM 435  
 SBSNAP 439  
 SCEERUN= parameter 106  
 SCHD= parameter 185, 404  
 SCHDTYP= parameter 66  
 SCIPROC= parameter 338  
 SCL= parameter 108  
 SCOPE= parameter 295  
 screen sizes, relating device names to 173  
 SECCNT= parameter 78, 109, 137  
 SECCOUNT= parameter 370, 374, 376  
 SECLVL= parameter 137  
 secondary master terminal  
   commands that can be copied to 73  
 SECUNIT= parameter 370, 374, 376  
 security  
   EXEC parameters 422, 426, 429  
   options 7  
 Security macro  
   description 2  
 SECURITY macro  
   COMM macro, and 136  
   description 135  
   IMSGEN macro, and 136  
   parameters 137  
     PASSWORD= 137  
     RCLASS= 137  
     SECCNT= 137  
     SECLVL= 137  
     TERMNL= 138  
     TRANCMD= 138  
     TYPE= 139  
   syntax diagram 135  
 SEGNO= parameter 186  
 SEGSIZE= parameter 172, 186, 456  
 selector pen  
   specifying in CONFIG macro 78  
 sequential buffering  
   CICS, and 472  
   control statements, specifying  
     SBCO 438  
     SBESNAP 440  
     SBIC 438  
     SBPARM 435  
     SBSNAP 439  
     SNAPDEST 441  
   online system 395  
   OSAM 430, 433  
 SERIAL= parameter 186  
 serviceability and trace options, defining 396  
 SESSION= parameter 128, 172  
 SETO 442  
 SETR 444  
 SGN= parameter 295  
 shared queues  
   interface, tracing 404  
 SHAREDQ= parameter 296  
 SHMSGSZ= parameter 297  
 SHUTDOWN= parameter 122  
 SIDE= parameter 415  
 SIGNON= parameter 346  
 SIMEXIT= parameter 78  
 single partition processing  
   HALDB 463  
 SIZE= parameter 172  
 SLDSDEF= parameter 375  
 SLU 1  
   TERMINAL macro statement, specifying 143  
 SLU 2  
   TERMINAL macro statement, specifying 143  
 SLU P  
   TERMINAL macro statement, specifying 143  
 SLU2= parameter 347  
 SMF  
   accounting procedures 6  
 SMP/E  
   LGEN definition 52  
   maintaining stage 1 output 52  
 SMP/E target zone (TLIB) 52  
 SMTO= parameter 347  
 SMTO1-8= parameter 347  
 SMTOG= parameter 347  
 SNAPDEST 441  
 SOD= parameter 297, 449  
 Sort/Split utility  
   LGEN Stage 1 processing 43  
 SOUT= parameter 297  
 SPA= parameter 187  
 SPACE= parameter 370, 375, 376  
 SPAP= parameter 297  
 SPAREA macro 140  
 specifying desired trace options 482  
 SPIE= parameter 297  
 SPM= parameter 298  
 Spool  
   TERMINAL macro statement, specifying 143

- SQGROUP= parameter 382
- SQTT= parameter 404
- SRCH= parameter 298
- SRMDEF= parameter 347, 461
- SSM= parameter 298
- SSN= parameter 476
- SST= parameter 476
- stage 1
  - binder order statements 50
  - ETO option 50
  - IMS.PROCLIB updates 327
  - input statements, entering 49
  - input, sequencing 41
  - LGEN processing 42
  - maintaining output with SMP/E 52
  - output 50
  - system definition, running 49
- stage 2
  - input 50
  - output 51
  - running 51
- START command
  - database access, defining 81
  - DATABASE macro, ACCESS parameter 81
  - dependent regions, starting 2
  - procedure, naming a 332
- START= parameter 406, 440
- static definition
  - terminals 25
  - when required 25
- STATION macro 140
  - description 140
  - parameters 140
    - ADDR= 140
    - ALIGN= 140
    - BACKUP= 141
    - FEAT= 141
    - NAME= 141
    - TRANSLM= 141
    - TYPE= 141
  - syntax diagram 140
- STIMER= parameter
  - message-driven programs 298
  - non-message driven programs 299
- STM= parameter 348
- STOP= parameter 406, 440
- storage estimates
  - extended private storage 47, 48
- storage pool definitions
  - default 377
  - DFSSPMxx, overriding defaults 377
- STRG= parameter 404
- STRINGMX= parameter 392
- STRINGNM= parameter 387
- Structured Call Interface (SCI) 450
- SUBPOOL macro
  - description 141
  - for switched communication devices 141
  - LU 6.1 VTAM Devices 142
  - parameters 142
    - MSGDEL= 142
- SUBPOOL macro (*continued*)
  - parameters (*continued*)
    - NAME= 142
  - syntax diagram 142
- SUBS= parameter 405, 482
- subsystem identification parameters 420
- SUF= parameter 299
- SUFFIX= parameter 103
- SURV= parameter 360
- SVC2= parameter 299
- SVCNO= parameter 86
- SVSODR= parameter 299
- SWAP= parameter 300
- SWITCH= parameter 361
- switched communication lines
  - described by LINE macro 111
- SYNCLEVEL= parameter 415
- Syntax Checker
  - checking parameter values 529
  - overview 332
- SYS= parameter 300
- SYS1= parameter 300
- SYS2= parameter 300
- SYSID= parameter 66, 125, 187
- SYSMSG= parameter 109
- system definition
  - ALL 29
  - BATCH 29
  - binder order statements 50
  - classes
    - DB/DC 29
    - DBCTL 29
    - DCCTL 29
  - CTLBLKS 29
  - examples
    - DB/DC environment 485
    - DC macro statements 490
  - IMSCTRL macro 87
  - macros 56
    - maximum occurrences 60
  - managing 41
  - MODBLKS 29
  - MSVERIFY 29
  - NUCLEUS 29
  - ON-LINE 29
  - preprocessor
    - description 43
    - executing 45
    - LGEN 47
    - parameters, defining 47
    - standard 47
    - storage requirements, estimating 47
    - when optional 43
  - procedures
    - environments in which they apply 192
  - RSR enabling 91
  - selecting appropriate 29
  - stage 1
    - input 49
    - output 50

system definition (*continued*)

- Stage 1
    - maintaining output with SMP/E 52
    - maintaining output with SMP/E, sample JCL 52
  - stage 1 input 49
  - stage 1 output 50
  - Stage 1 output
    - maintaining with SMP/E 52
    - maintaining with SMP/E, sample JCL 52
  - stage 1, running 49
  - stage 2
    - input 50
    - output 51
  - stage 2 input 50
  - stage 2 output 51
  - stage 2, running 51
  - storage requirements 48
  - tracking subsystem
    - when separate definition required 28
  - type
    - guidelines for selecting 97
  - types 29
    - used for most changes 30
  - verifying 43
  - when more than 4 MB of private storage is needed 43
- System Management Facility  
See see SMF
- System/3  
LINEGRP macro statement, specifying 114  
TERMINAL macro statement, specifying 143
- System/7  
LINEGRP macro statement, specifying 114  
TERMINAL macro statement, specifying 143
- SYSTEM= parameter 95

**T**

- TCORACF= parameter 300
- TERM= parameter 103
- terminal devices, allocating in an online system 194
- TERMINAL macro
  - COMM macro, and 78
  - description 143
  - label field 154
  - MFS 524
  - parameters 154
    - ADDR= 154
    - BACKUP= 155
    - BUFSIZE= 155
    - COMPT= 155
    - COMPTx= 156
    - EDIT= 159
    - FEAT= 159
    - FPBUF= 161
    - HEADCTL= 161
    - LTERM= 161
    - MODEL= 162
    - MODETBL= 162
    - MSGDEL= 163
    - NAME= 164

TERMINAL macro (*continued*)

- parameters (*continued*)
  - OPTIONS= 164
  - OUTBUT= 169
  - PTRSIZE= 171
  - PU= 171
  - SEGSIZE= 172
  - SESSION= 172
  - SIZE= 172
  - TYPE= 174
  - UNIT= 174
- syntax diagram 144
- terminals
  - defining 13
  - defining non-VTAM 14
  - dynamic 25
  - dynamic allocation 25
  - effect of system definition process on 25
  - static 25
  - static definition 25
- TERMNL= parameter 138
- TERMSEC= parameter 109
- TEST= parameter 301
- TIME= parameter 373
- TIMEOUT= parameter 449
- timer
  - counting unused IOVF control intervals 277
- TIMER= parameter 450
- TLIB 52
- TLIM= parameter 301
- TMINAME= parameter 98, 301, 376
- TPNAME= parameter 415
- TRACE command
  - DASD log activity, tracing 400
  - DB2 subsystem connection, tracing 405
  - DBF entries from FP, tracing 401
  - dispatcher activity, tracing 400
  - DL/I activity, tracing 400, 401
  - latch activity, tracing 402
  - lock activity, tracing 402
  - options 399
  - OTMA control, tracing 403
  - queue manager, tracing 403
  - RSR log router, tracing 402
  - scheduler, tracing 404
  - shared queues interface, tracing 404
  - storage manager calls, tracing 404
- trace entries
  - options, defining 396
  - storage requirements 405
- TRACE= parameter 301
- TRACK= parameter 301
- tracking subsystem
  - system definition
    - when separate definition required 28
- TRANCMD= parameter 138
- TRANSACT macro
  - description 174
  - parameters 177
    - AOI= 177
    - CODE= 178

TRANSACT macro *(continued)*parameters *(continued)*

DCLWA= 178  
 EDIT= 179  
 FPATH= 179  
 INQ= 180  
 INQUIRY= 180  
 MAXRGN= 180  
 MODE= 180  
 MSGTYPE= 181  
 PARLIM= 182  
 PROCLIM= 183  
 PRTY= 184  
 ROUTING= 185  
 SCHD= 185  
 SEGNO= 186  
 SEGSIZE= 186  
 SERIAL= 186  
 SPA= 187  
 SYSID= 187  
 WFI= 188

syntax diagram 175

TRANSLM= parameter 141

transport manager subsystem

installing 323

VTAM definition 323

TRKMODS= parameter 377

TRN= parameter 301

TRUNC= parameter 348

TSR= parameter 302

tuning considerations

control program address spaces 7

TYPE macro

description 188

MFS 524

parameters 189

BACKUP= 189

EDIT= 189

OPTIONS= 189

UNITYPE= 190

syntax diagram 188

TYPE= parameter 129, 131, 139, 141, 174, 481

types

system definition 29

**U**

UHASH= parameter 302

UHTS= parameter 302

UJCL1= parameter 108

UJCL2= parameter 108

UJCL3= parameter 108

UJCL4= parameter 108

UJCL5= parameter 108

UJCLx= parameter 108

UMAC0= parameter 106

UMAC1= parameter 106

UMAC2= parameter 106

UMAC3= parameter 106

UNIT= parameter 174, 370, 374, 375, 456

UNITYPE= parameter 117, 190, 456

UPDTPRT= parameter 104

USERID= parameter 450

USERLIB= parameter 106

USERVAR= parameter 302, 361

USRHASH= parameter 84

**V**

VACBOPN= parameter 349

VALCK= parameter 302

VAUT= parameter 302

VERIFY= parameter 113

VFREE= parameter 303

VSAM

buffer pools, defining 385, 430

performance options, defining 391

subpools, defining 387, 432

VSAMFIX= parameter 392

VSAMPLS= parameter 392

VSFX= parameter 303

VSPEC= parameter 303

VTAM

COMM macro, and 70, 72, 76

data communication macros

COMM 20

NAME 20

SUBPOOL 20

TERMINAL 20

TYPE 20

VTAMPOOL 20

transport manager system definition 323

using multi-node persistent sessions (MNPS) to take over terminals 357

VTAM terminals

dynamic 25

dynamic allocation 25

effect of system definition process on 25

static 25

static definition 25

using multi-node persistent sessions (MNPS) to take over 344

when multi-node persistent sessions (MNPS) are used rather than XRF 72

VTAMPOOL macro 190

**W**

WADS= parameter 303, 408, 421

WADSDEF= parameter 408

WFI= parameter 188

WKAP= parameter 303

**X**

XPLINK= parameter 303

XRF

COMM macro

APPLID parameter 72

PASSWD parameter 76

CTLUNIT macro, BACKUP parameter 80

DFSHSBxx PROCLIB member 357

XRF (*continued*)  
system definition 41

## Y

YEAR4= parameter 303

## Z

z/OS

IMS procedures, and 433

IMSRDR procedures, and 433

z/OS application programs

accessing IMS databases using ODBA (Open  
Database Access) 465

accessing IMS databases using Open Database  
Access (OBDA) 465





Program Number: 5655-J38

IBM Confidential  
Printed in USA

ZES1-2349-02





Spine information:



IMS

Installation Volume 2: System Definition and Tailoring Version 9