System Automation for OS/390

AOC/MVS IMS Automation Programmer's Reference and Installation Guide

Version 1 Release 4

SC23-3817-02

System Automation for OS/390

AOC/MVS IMS Automation Programmer's Reference and Installation Guide

Version 1 Release 4

SC23-3817-02

Note

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

Third Edition (June 1999)

This edition applies to Version 1 Release 4 of AOC/MVS IMS Automation Feature (5685-151), and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

A form for readers’ comments appears at the back of this publication. If the form has been removed, address your comments to:

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

FAX: (Germany) 07031-16-3456
FAX: (Other countries) (+49)+7031-16-3456

IBM Mail Exchange: DEIBMBM9 at IBMMAIL
Internet: s390id@de.ibm.com

Contents

Notices . . . xiii
Programming Interface Information . . . xiii
Trademarks . . . xiii

About This Book . . . xv
About IMS Automation . . . xv
What IMS Automation Does . . . xv
Customized IMS Automation . . . xvi
Who Should Use This Book? . . . xvi
What Is In This Book? . . . xvi
What Is In IMS Automation? . . . xvii
What's New in This Book? . . . xvii
Related Publications . . . xviii
The SA OS/390 Library . . . xviii
Related Product Information for the Base Program . . . xix
Related Product Information for Workstation Operations . . . xx

Part 1. Understanding the IMS Automation Feature . . . 1

Chapter 1. Components of Automation . . . 3
NetView Automation Table . . . 4
Automation Operators . . . 4
Control File and Status File . . . 5
Control File . . . 5
Status File . . . 6
State/Action Tables . . . 7
Program-to-Program Interface . . . 8
AOI Exit . . . 8
Operator Interface and Single-Point-of-Control . . . 8
Controlling and Monitoring Multiple Domains . . . 9
Message Handling . . . 9
Critical Messages Manager . . . 10
Operator Interface Panels . . . 11
Processing Environment . . . 13
Functional Overview . . . 14
Startup Overview . . . 14
Shutdown Overview . . . 15
Recovery Overview . . . 16
Automatic Restart Manager (ARM) . . . 17
Takeover Overview . . . 18
XRF Takeover . . . 18
XRF and ARM Takeover . . . 21
| FDR Environment . . . 21
| Functional Overview . . . 21
| Startup Overview . . . 21
| Recovery Overview . . . 21
| Shutdown Overview . . . 22

Chapter 2. How to Implement and Use IMS Automation . . . 23

Using Single-Point-of-Control . . . 24
Added Capabilities . . . 24
How Single-Point-of-Control Works . . . 24
SPOC Requirements . . . 25
Setting Up SDF for IMS Automation . . . 25
IMS Automation SDF Panels . . . 27
Coding SDF—A Basic Overview . . . 29
Defining IMS Automation Startups and Shutdowns . . . 30
Grouping IMS . . . 30
Parent/Child Relationships in a Startup or Shutdown . . . 30
Defining the Service Periods . . . 31
Defining Startup and Shutdown Triggers . . . 32
IMSPOST Command Processor . . . 32
Defining Triggers in the Control File . . . 32
Posting Triggers . . . 32
Using Triggers in a Distributed Environment . . . 33
Defining Transaction Recovery . . . 34
Using State/Action Tables . . . 36
State/Action Table Control File Entries . . . 36
Example Control File Entries for State/Action Table Definitions . . . 37
State/Action Table Keywords . . . 37
Maintaining Security . . . 38

Part 2. Programming the IMS Automation Feature . . . 39

Chapter 3. Programming Information . . . 41
Differences in AOC/MVS Release 1 and SA OS/390 1.3 Syntax . . . 41
Upgrading from IMS Automation 1.3 to IMS Automation 1.4 Using ARM Support 41
Special Considerations when Coding Control File Entries . . . 41
Restrictions . . . 42
Using Multiple Control File Members for Configuration Data . . . 42
Changing Control File Entries . . . 42
Using Partial Reload for Control File Entries . . . 43
Using a Master Control File . . . 43
Notational Conventions and Syntax Rules . . . 44
ABCODEPROG—Respond to BMP region abends . . . 45
ABCODES/ACTCODES/ALTCODES—Restart control region after abend . . . 47
ABCODETRAN—Respond to transaction abend codes . . . 49
ACORESTART—SA OS/390 . . . 51
AREA—Define a set of state/action tables . . . 52
AUTOOPS—Define global automation tasks to SA OS/390 . . . 53
AUTOOPS—Define automation tasks for IMS subsystems . . . 54
BRO—Broadcast a message prior to shutdown . . . 55
CHE—Issue a checkpoint command . . . 56
| CQS0031A—Confirm CQS restart for structure . . . 57
| CQS0032A—Respond to CQS structure restart . . . 58
| CQS0033A—Respond to client takeover restart . . . 59
| CQSET—Issue Structure Checkpoint at CQS Termination . . . 60
DFS2142—Respond to stopped logical link path message . . . 61
DFS2161I—Link stopped by other system . . . 63
DFS2169I—Respond to MSC link disconnection message . . . 64
DFS3258A—No online data sets available . . . 65
DFS3869A/DFS3869I—Alternate XRF IMS subsystem detects potential failure 66

DFS554A—Respond to program abend 67
DFS810A—Define restart commands 69
DFS989I—Define Restart Commands (DBCTL Only) 71
DFS994I—Respond to checkpoint written to the IMS log 73
ENVIRON—Tailor operations 76
ENVIRON SETUP—Identify startup routine 83
EXTCOND—External conditions 84
FORCE—Takeover by alternate XRF IMS subsystem 86
HOLDQ—Issue commands at shutdown 87
IMSCNTL—IMS/DC control region 88
IMS CRITMSGS—Display critical messages in the SDF 90
IMSGROUP—Group of IMS subsystems 91
IMSRSENM—IMS resource name 92
INITSTART—Control subsystem start 93
IPS—XRF IMS takeover commands 95
NTFYOP—Define notification operator 96
OLDS—Define recovery criteria for OLDS 97
POSTCHKP—Issue commands after IMS shutdown checkpoint 100
PRECHKP—Issue commands prior to IMS shutdown checkpoint 101
PRODUCT—Subsystem state/action table sets 102
RECONS—Set monitoring interval for RECONS 103
RECOVERY—Turn automation on/off 104
RELEASEQ—Issue commands after shutdown completes 106
RESTART—Control restart of IMS subsystem 107
RESTARTABORT—Emergency restart commands 108
SDF support 110
SERVICE—Service period definitions 112
SHUTINIT—Initial shutdown commands 114
SHUTNORM/SHUTIMMED/SHUTFORCE—Control region 115
SHUTNORM/SHUTIMMED/SHUTFORCE—Message, FP, and BMP regions 117
SHUTTYPES—Issue commands at operator shutdown 120
SNAPQ—Issue SNAPQ checkpoint command 123
STARTn—Define start commands 124
STARTUP—Issue commands at IMS subsystem startup 126
STOPBMPREGION—Stop batch message regions 128
STOPFPREGION—Stop fast path regions 129
STOPREGION—Stop IMS dependent message region 130
STSFILE—Status file 131
SUBSYSTEM—Define IMS region to SA OS/390 133
TCO—Issue commands for time-driven procedures 136
TCOMEMBERS—Define TCO Members 137
THRESHOLDS—IMS error thresholds 139
TPABEND—Dependent region abend 141
TRIGGER—Startup and shutdown triggers 142
UNLKAVM—Reply to message AVM005A 144
UNLOCK—Input/output prevention completed 145
| UP—Initiate FDR Startup 146
USERSTART—User-defined restart commands 147
VTAMTERMS—Release terminal sessions 148
%INCLUDE—Include additional members in control file 149
$PI—Stop BMP tasks on active subsystem 150
$SI—Issue start initiator commands after XRF takeover 151
$TI—Assign job classes after XRF takeover 152

Chapter 4. Common Routines . . . 153
EVIEX002—Retrieve XRF partner data . . . 154
EVIEX003—Update status file data . . . 156
EVIEX004—Retrieve XRF partner data . . . 163
IMSBMSG—Build message processor . . . 165
IMSCMD—Issue IMS commands . . . 167
IMSFWM—Add messages to SDF . . . 170
IMSPOST—Post an external event . . . 172
IMSQRY—Name lookup . . . 174
IMSRCMD—Request an IMS function . . . 178
IMSSEC—Invoke security checking . . . 179
EVISROUT—IMS message router . . . 181
EVISROUT—IMS message router (contd.) . . . 182
Adding Local Applications to the IMS Automation Operator Interface . . . 182

Part 3. Planning, Installing, and Customizing the IMS Automation Feature . . . 185

Chapter 5. Planning . . . 187
Pre-Installation and Planning Steps . . . 187
Designing Your Automation Environment . . . 188

Chapter 6. Installation . . . 189
Step 1: Load the Tape Files . . . 189
Step 2: Edit MPFLSTxx . . . 189
Step 3: Update IEAAPFxx in SYS1.PARMLIB . . . 190
Step 4: Define Subsystem Allocatable Consoles . . . 190
Step 5: Check the Subsystem Name Table (IEFSSNxx) . . . 191

Chapter 7. Merge NetView Related Members . . . 193
Step 1: Add IMS Automation Data Sets to NetView JCL . . . 193
Step 2: Copy IMS Automation Sample Members to the Target Library . . . 194
Step 3: Merge SDF Members . . . 194
Step 4: Include IMS Control File Entries . . . 194
Step 5: Include Required Command Model Statements into DSICMD . . . 194
Step 6: Include and Update the Automation Table . . . 195
Step 7: Include Operator Definitions into DSIOPF . . . 196
Step 8: Add Additional VTAM APPL Statements . . . 196
Step 9: Merge or Include the NetView Profile Data Set . . . 196
Step 10: Include Task Statements into DSIDMN and Update . . . 196

Chapter 8. IMS Automation IMS Definitions . . . 197
Step 1: Modify and Run the IMS SYSGEN . . . 197
Step 2: Define IMS PSB Entries . . . 198
Step 3: Define IMS Security Gen Entries . . . 198
Step 4: Define IMS BMP Procedure . . . 199
Step 5: Create PPI Initialization Member for IMS . . . 199
Step 6: Schedule Each IMS to be Stopped and Restarted . . . 199

Chapter 9. IMS Automation Definitions in NetView . . . 201
Step 1: Modify the SA OS/390 ENVIRON Base Control File Entry . . . 201
Step 2: Add the Automation Operators . . . 202
Step 3: Add the Automation Class to the Notification Operator Definitions . . . 202
Step 4: Code the Entries for Control Region . . . 203

Step 5: Code the Entries for DBRC/DLISAS Regions 205
| Step 6: Code the Entries for FDR Region 206
| Step 7: Code the Entries for CQS Region 206
Step 8: Code the Entries for Message Regions 206
Step 9: Code the Entries for OLDS 207
Step 10: Code the Entries for MSC Links 208
Step 11: Code the Entries for RECONs Recovery 208
Step 12: Code the Entries for Transaction and Program Recovery 208
Step 13: IMS Automation Security Checking 209
Step 14: Extended IMS Automation Common Control File Definitions 210
Step 15: Preparing IMS Automation to Manage an IMS XRF System in a Dual-CPC Environment 210
Step 16: XRF Considerations in an ARM Environment 212
Step 17: REXX Environment Table 213
Step 18: Customizing the SDF 214

Chapter 10. Optional Additions to the PPI 217
Step 1: Extend EVISPINM Member, If Required 217
Step 2: Extend EVINTASK, If Required 222

Chapter 11. Testing IMS Automation 225
Startup Test Scenario 226
Message Region Test Scenario 227
Shutdown Test Scenario 228

Part 4. Appendixes 229

Appendix A. MVS Sample JCL 231
Copy IMS Automation Feature AEVISAMP Members to NetView DSIPARM Data Set 232
EVISJ010 232
Assemble and Link-Edit EVISPINM Tables 234
EVISJ020 234

Appendix B. Automation and Productivity Enhancements - Examples . 237
Required NetView Automation Table Entries 239
EVISPINM 239
EVINTASK 243
IMSMSG00 (EVIMSG00) 245
EVIMCON0 246
EVIMEVI0 247
IMSMSG01 (EVIMSG01) 248
EVIMAVM1 250
EVIMCON1 251
EVIMDFS1 253
EVIMDSP1 262
EVIMDXR1 263
| EVIMCQS1 264
EVIMEVI1 266
EVIMIOS1 268
NetView Operator Definition 269
EVIOPF 269
Required Command Model Entries 270

EVICMD . . . 270
EVISECUR . . . 273
Sample Configuration Tables . . . 274
EVIDMN . . . 274
EVISTSM . . . 275
EVIS2CFG . . . 276
EVIS2STA . . . 277
EVIS2OPS . . . 278
EVIS2IMS . . . 279
EVIS2DBC . . . 288
EVIS2XRF . . . 295
| EVIS2SPX . . . 312
| EVIS2FDR . . . 322
| EVIS2CQS . . . 323
| EVIS2CRT . . . 325
EVIS2LCL . . . 326
EVIPRFAO . . . 329

Appendix C. DB2 in an XRF Environment . . . 331
Takeover Overview—Example 1 . . . 332
Conclusions of Example 1 . . . 332
Takeover Overview—Example 1 (contd.) . . . 332
Takeover Overview—Example 1 (contd.) . . . 334
Takeover Overview—Example 1 (contd.) . . . 336
Normal Takeover Sequence—Example 2 . . . 338
Conclusions of Example 2 . . . 338
Normal Takeover Sequence—Example 2 . . . 338
Normal Takeover Sequence—Example 2 (contd.) . . . 340
Normal Takeover Sequence—Example 2 (contd.) . . . 342
Normal Takeover Sequence—Example 2 (contd.) . . . 344
Forced Takeover, Cancel IRLM/IMS—Example 3 . . . 346
Conclusions of Example 3 . . . 346
Forced Takeover, Cancel IRLM/IMS—Example 3 (contd.) . . . 347
Forced Takeover, Cancel IRLM/IMS—Example 3 (contd.) . . . 349
Forced Takeover, Cancel IRLM/IMS—Example 3 (contd.) . . . 351
Forced Takeover, Cancel IRLM/IMS—Example 3 (contd.) . . . 353
Forced Takeover, Cancel IMS/IRLM—Example 4 . . . 355
Conclusions of Example 4 . . . 355
Forced Takeover, Cancel IMS/IRLM (contd.) . . . 357
Forced Takeover, Cancel IMS/IRLM (contd.) . . . 359

Appendix D. Installation Worksheets . . . 363
Planning Worksheets . . . 364
NetView Domains and IMS Regions . . . 364
IMS Subsystem Information Sheets . . . 365
Automation Operator Information Sheets . . . 366
IMS Subsystem Group Information Sheet . . . 367
Service Period Information Sheets . . . 368
Triggers Worksheet . . . 369
System IPLs, NetView Restarts, and IMS Restarts . . . 370
Customization Worksheets . . . 372
SA OS/390 Requirement Worksheet . . . 372

Glossary of IMS Automation Terms . . . 373

Index 377

Figures

1. Sample State/Action Table . . . 7
2. SDF Sample Panel . . . 10
3. IMS Automation in a NetView environment . . . 14
4. Start Functions Panel . . . 15
5. Operator Shutdown Panel . . . 16
6. XRF Complex Before Takeover . . . 19
7. XRF Complex During Takeover . . . 20
8. XRF Complex After Takeover . . . 20
9. Focal Point Definitions in a Distributed Environment . . . 24
10. Tree Structures of a Distributed Environment . . . 26
11. Data Center Systems Panel—Sample . . . 27
12. IMS Monitor Panel . . . 28
13. Detail Display of Critical Message . . . 29
14. Master Control File Example . . . 43
15. IMS Automation: Operator Shutdown Panel . . . 122
16. Options Menu Available from Operator Shutdown Panel . . . 122
17. EVISPINM PPI Initialization Member . . . 219
18. EVINTASK PPI Initialization Member . . . 222
19. MVS System with Test and Production NetView . . . 224
20. DB2/XRF Takeover Overview . . . 333
21. DB2/XRF Takeover Overview (contd.) . . . 335
22. DB2/XRF Takeover Overview (contd.) . . . 337
23. DB2/XRF Normal Takeover Sequence . . . 339
24. DB2/XRF Normal Takeover Sequence (contd.) . . . 341
25. DB2/XRF Normal Takeover Sequence (contd.) . . . 343
26. DB2/XRF Normal Takeover Sequence (contd.) . . . 345
27. DB2/XRF Forced Takeover, Cancel IRLM/IMS . . . 348
28. DB2/XRF Forced Takeover, Cancel IRLM/IMS (contd.) . . . 350
29. DB2/XRF Forced Takeover, Cancel IRLM/IMS (contd.) . . . 352
30. DB2/XRF Forced Takeover, Cancel IRLM/IMS (contd.) . . . 354
31. DB2/XRF Forced Takeover, Cancel IMS/IRLM . . . 356
32. DB2/XRF Forced Takeover, Cancel IMS/IRLM (contd.) . . . 358
33. DB2/XRF Forced Takeover, Cancel IMS/IRLM (contd.) . . . 361

Notices

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

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 Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

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

Programming Interface Information

This book documents programming interfaces that allow the customer to write programs to obtain the services of System Automation for OS/390.

Trademarks

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

AT	IPDS	RMF
CICS	MVS/ESA	SP
DB2	MVS/XA	System/370
IBM	NetView	System/390
IMS	OS/390	VTAM
IMS/ESA	RACF	

About This Book

This book describes how to install, customize, and maintain the AOC/MVS IMS Automation Feature, hereafter referred to as SA OS/390 IMS Automation Feature or simply IMS Automation. This is due to the fact that AOC/MVS 1.4 has been withdrawn and replaced by System Automation for OS/390 (SA OS/390).

About IMS Automation

IMS Automation provides a single point of control for IMS startup, shutdown, display, recovery, XRF takeover and Fast Database Recovery (FDR) operations, based on the automation environment supported by System Automation for OS/390. IMS Automation provides new functions that are not available in NetView, IMS, or SA OS/390, resulting in a more comprehensive automation capability than is possible with these products individually. It offers synergy for the separate capabilities of IMS and NetView, incorporating the strengths of each product into a very effective operations manager for IMS processing activities.

The benefits of IMS Automation are multiplied in an XRF IMS environment where the purpose is to maintain an alternate IMS subsystem. In such an environment, IMS Automation can transfer workload to another set of available resources (takeover) quickly and with minimal disruption. The end user benefits through the significantly reduced impact of IMS outages, whether scheduled or unscheduled. Overall benefits include enhanced operator productivity and reduced error potential.

What IMS Automation Does

IMS Automation provides automation functions in the following four categories:

Startup
: The IMS Automation startup function enables the operator to start an IMS subsystem according to predefined procedures in either an XRF, non-XRF, or FDR environment. In addition, the IMS Automation operator can bypass these user-defined start procedures and manually start IMS subsystems.

Shutdown
: The IMS Automation shutdown function allows the operator to select from three shutdown types:

- NORMAL
- IMMEDIATE
- FORCE

Operators can abort an in-flight shutdown, broadcast a message informing users of a pending shutdown, and specify an immediate restart of the subsystem once shutdown is complete.

Recovery
: The IMS Automation recovery function allows IMS subsystem recovery in an XRF or FDR environment or restart in a non-XRF environment. IMS Automation analyzes errors detected during startup and shutdown processes. From this analysis, IMS Automation will initiate automated recovery actions or notify the operator of required action. IMS Automation attempts automatic recovery for IMS region failures, application transaction abends, archiving

errors, OLDS failures, and MSC link failures. When automated recovery is not possible, the operator is advised about the manual recovery actions required.

Takeover
The IMS Automation takeover function assists the transfer of resources from a failing active IMS subsystem to its alternate IMS subsystem in an XRF environment.

Customized IMS Automation

IMS Automation is customized in the same way that other SA OS/390 options are customized. A system programmer codes control file entries and modifies the IMS and SDF parameter members to specify how the IMS functions are to perform.

Refer to Chapter 9, “IMS Automation Definitions in NetView” on page 201 for the entries that may be modified.

Who Should Use This Book?

This book is written for the following audiences:

Evaluators of IMS Automation
Those users who evaluate productivity and high availability tools for IMS to determine whether to purchase IMS Automation. These users are assumed to have a basic understanding of NetView and the IMS operations environment.

Programmers
The system programmers identified and scheduled for training who:

- Understand MVS console operations, concepts, and system programming
- Understand IMS console operations and system programming
- Have attended prior SA OS/390 training

What Is In This Book?

This book is divided into three parts:

- Part 1, Understanding the IMS Automation Feature, provides a broad perspective that you will need to understand how to program IMS Automation.
- Part 2, Programming the IMS Automation Feature, provides programming syntax, language elements, and conventions that you will need to understand before you install and customize IMS Automation.
- Part 3, Planning, Installing, and Customizing the IMS Automation Feature, provides sequential instructions for installing, modifying, customizing, and testing IMS Automation.

In addition to the main text, this book contains the following reference materials:

- Appendix A, MVS Sample JCL, contains an installation JCL for this product.
- Appendix B, Automation and Productivity Enhancements - Examples, includes parameter files used to customize IMS Automation.
- Appendix C, DB2 in an XRF Environment, describes XRF IMS DB2 examples.

- Appendix D, Installation Worksheets, provides the IMS Automation installation worksheets.

What Is In IMS Automation?

With IMS Automation, operators can:

| - Perform automated startup, shutdown, XRF takeover, and FDR recovery
- Display status of subsystems, active regions, and shutdown status
- Start message and control regions
- Start, stop, and display IMS message regions
- Schedule regular startups and shutdowns with service periods
- Define initialization conditions with startup and shutdown triggers
- Perform TCO and DBCTL functions from the IMS Automation interface
- Display and obtain information about critical messages
- Broadcast messages to specific subsystems
- Perform support functions, including setting traces and working with the program-to-program interface

IMS Automation functions support:

- Subsystems other than IMS related subsystems
- Fast path regions
- BMP regions with their own outstanding WTOR
- Dependent regions defined as regular subsystems

| - Common Queue Server (CQS) regions

| - Fast Database Recovery (FDR) regions

- IMS Automation supplied automation operator exits calling user-defined automation operations exits

What's New in This Book?

Changes to this book since Release 3 are:

- Modifications supporting Release 4 of IMS Automation
- IMS Automation support of Release 4 of SA OS/390 and MVS/ESA SP Version 5.2 automatic restart manager (ARM), which allows a subsystem to be restarted on another system image of the sysplex. This is available in the SA OS/390 IMS automatic restart manager special program enhancement (SPE).
- Modifications relating to the enhancements introduced by APAR OW32833 and OW35606 which provide IMS V6 toleration and new function support, namely FDR and CQS environment support.
- Technical changes reflecting service updates

A vertical bar (|) in the left margin indicates changes to the text and illustrations since the last edition.

Related Publications

The SA OS/390 Library

The following table shows the information units in the SA OS/390 library:

Table 1. SA OS/390 Library

Title	Order Number
SA OS/390 General Information	GC28-1541
SA OS/390 Licensed Program Specifications	GC28-1540
SA OS/390 Planning and Installation	GC28-1549
SA OS/390 Customization	GC28-1566
SA OS/390 Operations	GC28-1550
SA OS/390 Messages and Codes	GC28-1569
SA OS/390 Technical Reference	GC28-1593
AOC/MVS CICS Automation General Information	GC23-3813
AOC/MVS CICS Automation Operator's Guide	SC23-3815
AOC/MVS CICS Automation Programmer's Reference and Installation Guide	SC23-3814
AOC/MVS IMS Automation General Information	GC23-3816
AOC/MVS IMS Automation Operator's Guide	SC23-3818
AOC/MVS IMS Automation Programmer's Reference and Installation Guide	SC23-3817
AOC/MVS OPC Automation General Information	GC23-3819
AOC/MVS OPC Automation Operator's Guide and Scheduler's Reference	SC23-3821
AOC/MVS OPC Automation Programmer's Reference and Installation Guide	SC23-3820

The System Automation for OS/390 books (except Licensed Program Specifications) are also available on CD-ROM as part of the following collection kits:

- IBM Online Library OS/390 Collection (SK2T-6700)
- IBM Online Library Networking Collection (SK2T-6012)

These softcopy collections include the IBM Library Reader, a program that enables you to view online documentation.

SA OS/390 Homepage

For the latest news on SA OS/390, visit the SA OS/390 homepage at http://www.s390.ibm.com/products/sa/

Related Product Information for the Base Program

The following table shows the books in the related product libraries that you may find useful for support of the SA OS/390 base program.

Table 2 (Page 1 of 2). Related Product Books

Title	Order Number
MVS/ESA MVS Configuration Program Guide and Reference	GC28-1817
MVS/ESA Planning: Dynamic I/O Configuration	GC28-1674
MVS/ESA Support for the Enterprise Systems Connection	GC28-1140
MVS/ESA Planning: APPC Management	GC28-1110
MVS/ESA Application Development Macro Reference	GC28-1822
MVS/ESA SP V5 System Commands	GC28-1442
MVS/ESA SPL Application Development Macro Reference	GC28-1857
NetView for MVS V3R1 Administration and Security Reference	SC31-8045
NetView for MVS V3R1 Automation Implementation	SC31-8050
NetView for MVS V3R1 Automation Planning	SC31-8051
NetView for MVS V3R1 Command Reference	SC31-8047
NetView for MVS V3R1 Customization Guide	SC31-8052
NetView for MVS V3R1 Customization: Writing Command Lists	SC31-8055
NetView for MVS V3R1 Installation and Administration Guide	SC31-8043
NetView for MVS V3R1 RODM and GMFHS Programming Guide	SC31-8049
NetView for MVS V3R1 User's Guide	SC31-8056
NetView for MVS V3R1 Tuning Guide	SC31-8048
OS/390 Hardware Configuration Definition: User's Guide	SC28-1848
OS/390 Information Roadmap	GC28-1727
OS/390 Information Transformation	GC28-1985
OS/390 Introduction and Release Guide	GC28-1725
OS/390 V1R2.0 JES Commands Summary	GX22-0041
OS/390 Licensed Program Specifications	GC28-1728
OS/390 Printing Softcopy Books	S544-5354
OS/390 Starting Up a Sysplex	GC28-1779
OS/390 Up and Running!	GC28-1726
Planning for the 9032 Model 3 and 9033 Enterprise Systems Connection Director	SA26-6100
Resource Access Control Facility (RACF) Command Language Reference	SC28-0733
S/390 MVS Sysplex Overview – An Introduction to Data Sharing and Parallelism	GC23-1208
S/390 MVS Sysplex Systems Management	GC23-1209
S/390 Sysplex Hardware and Software Migration	GC23-1210
S/390 MVS Sysplex Application Migration	GC23-1211

Table 2 (Page 2 of 2). Related Product Books	
Title	**Order Number**
S/390 Managing Your Processors	GC38-0452
TSO/E REXX/MVS Users Guide	SC28-1882
TSO/E REXX/MVS Reference	SC28-1883
Using the Enterprise Systems Connection Manager	SC23-0425
VSE/SP Unattended Node Support	SC33-6412
VSE/ESA 1.1.0 Unattended Node Support	SC33-6512
VTAM Version 3 Release 3 Network Implementation Guide	SC31-6404
VTAM Version 3 Release 4 Network Implementation Guide	SC31-6434

Related Product Information for Workstation Operations

The following are the books in the related product libraries that you may find useful for support of SA OS/390 workstation operations.

Table 3. Related Product Books	
Title	**Order Number**
APPC System Definitions in MVS/ESA and OS/2	GG66-3224
APPC Programming Considerations	GG24-3818
APPC Application Examples	GG24-3819
Distributed Console Access Facility User's Guide	GE13-0061
IBM Communications Manager/2 Version 1.1	G221-3630
IBM Communications Manager/2 Version 1.1 Information and Planning Guide	SC31-7007
IBM Communications Manager/2 Version 1.1 Workstation Installation Guide	SC31-6169
IBM Communications Manager/2 Version 1.1 Configuration Guide	SC31-6171
IBM Communications Manager/2 Version 1.1 User's Guide	SC31-6108
IBM Operating System/2 Version 2.1 Using the Operating System	S61G-0905
IBM Operating System/2 Warp	SR28-5668
NetView for MVS V3R1 Graphic Monitor Facility User's Guide	SC31-8095
Official Guide to Using OS/2 Warp	SR28-5659
Personal Communications Programmer's Guide	SC31-8660
Personal Communications Reference	SC31-8259
Personal Communications Tell Me About OS/2 Access Feature	SC31-8257
Personal Communications Up and Running	SC31-8258

Part 1. Understanding the IMS Automation Feature

Chapter 1. Components of Automation . . . 3
NetView Automation Table . . . 4
Automation Operators . . . 4
Control File and Status File . . . 5
  Control File . . . 5
  Status File . . . 6
State/Action Tables . . . 7
Program-to-Program Interface . . . 8
AOI Exit . . . 8
Operator Interface and Single-Point-of-Control . . . 8
  Controlling and Monitoring Multiple Domains . . . 9
  Message Handling . . . 9
  Critical Messages Manager . . . 10
  Operator Interface Panels . . . 11
Processing Environment . . . 13
Functional Overview . . . 14
  Startup Overview . . . 14
  Shutdown Overview . . . 15
  Recovery Overview . . . 16
  Takeover Overview . . . 18
| FDR Environment . . . 21
| Functional Overview . . . 21

Chapter 2. How to Implement and Use IMS Automation . . . 23
Using Single-Point-of-Control . . . 24
  Added Capabilities . . . 24
  How Single-Point-of-Control Works . . . 24
  SPOC Requirements . . . 25
Setting Up SDF for IMS Automation . . . 25
  IMS Automation SDF Panels . . . 27
  Coding SDF—A Basic Overview . . . 29
Defining IMS Automation Startups and Shutdowns . . . 30
  Grouping IMS . . . 30
  Parent/Child Relationships in a Startup or Shutdown . . . 30
  Defining the Service Periods . . . 31
  Defining Startup and Shutdown Triggers . . . 32
  Using Triggers in a Distributed Environment . . . 33
Defining Transaction Recovery . . . 34
Using State/Action Tables . . . 36
  State/Action Table Control File Entries . . . 36
  State/Action Table Keywords . . . 37
Maintaining Security . . . 38

Chapter 1. Components of Automation

SA OS/390 provides most of the components required to automate your system resources. The IMS Automation Feature, an extension of SA OS/390, meets the needs of the complex IMS environment by providing the following additional functions:

- Automated startup and shutdown processes, including the defined dependent components (DBRC, DLI, MPR, IFP, BMP) and CQS and FDR regions.
- Single-point-of-control capability, allowing you to monitor and control all IMS regions in your organization, local and remote, from one NetView operator session
- Fully automated XRF takeover or FDR recovery
- Monitoring of critical IMS resources, such as application transactions and MSC links

The following list describes some of the automation components used with SA OS/390 and IMS Automation. For a complete and more comprehensive list of automation components, refer to the SA OS/390 base documentation.

NetView
NetView is the environment in which automation routines are controlled and invoked. Several of the NetView components are used by our automation routines, such as the:

- NetView automation table
- Command Model Statements member
- Operator Definition member

The NetView automation table receives console messages and calls automation routines based on those messages.

Control file
The SA OS/390 control file defines the:

- Environment where automation will occur
- Resources to be automated
- Extent and characteristics of automation

Status file
The SA OS/390 status file keeps a log of startup and shutdown information, such as:

- When a subsystem is initialized
- When a subsystem is shut down
- Service periods

Automation operators
Automation operators execute actions similar to human operators through defined procedures and actions specified in the control file.

Operator interface and single-point-of-control
A comprehensive set of panels are provided that allow the operator to monitor and control IMS subsystems from a NetView console. When single-point-of-control is used, all IMS subsystems, local and remote, can be controlled from one operator session on the focal point.

Single-point-of-control uses the NetView-to-NetView task (NNT) as the interface to other NetView domains.

Program-to-program interface
IMS Automation uses NetView's program-to-program interface to enable users to send and receive data from other programs. There is a NetView side and an IMS side to the program-to-program interface.

State/Action table
State/Action tables invoke automation on the basis of event and status relationships.

AOI exit
IMS Automation uses the IMS automation operator exit, DFSAOUE0, to trap messages destined for the IMS MTO operator. These messages can then be routed to NetView, suppressed from the MTO console, or sent to the MVS system console.

NetView Automation Table

Whenever NetView receives a message, it scans the automation table to see if actions are required as a result of the message. Consider the following example:

```
IF MSGID  = '123'
   THEN EXEC( CMD('EVIXZ123 '));
```

This states, "When message **123** is received, execute **EVIXZ123**." EVIXZ123 verifies that the region that issued the message is defined in the control file and then takes the automated actions defined for this message.

Similarly, IMS Automation automates many processes from this table. Though this is a simple example of a automation table entry, the automation table is a very powerful automation component. Through the automation table, IMS Automation can respond immediately and consistently to messages. The automation table also enables IMS Automation to account for differences between different IMS versions. As a result, IMS Automation prevents a delay in response to events which could affect integrity or availability.

Messages may be user-defined as critical in the NetView automation table, causing them to display in the Critical Messages Manager. Messages display in the Critical Messages Manager in a prioritized, scrollable format. For more information, see "Critical Messages Manager" on page 10.

IMS Automation provides an extension to the automation table that defines default automation responses to IMS messages.

Automation Operators

Automation operators execute actions similar to a human operator by responding to operating system, subsystem, and network messages. Each automation operator runs as a separate NetView task. This task design allows for concurrent, parallel processing. It also permits NetView to distribute its workload among the automation operators, thereby improving system performance. Following is a list of keywords used to define the automation operators unique to IMS Automation:

IMSMSTR Identifies the automation operator that provides message distribution services for IMS automation. The message content is used to identify the region for which this message applies. Messages are then routed to the automation operators defined with the IMSOP*nn* keywords, which are region specific.

IMSOP*nn* Identifies the automation operators that handle messages for specific IMS regions.

IMSPPI Handles program-to-program interface communications for IMS.

IMSWATCH Handles the XRF surveillance monitoring during a takeover situation.

Control File and Status File

The control file contains the automation policy information for each subsystem, such as settings for automation flags and IMS subsystem definitions. The status file contains the current operational status for all automated resources. IMS Automation uses an extended command processor to update the status for a specific IMS subsystem. The automation policy and resource status determine what automated actions are performed and under what conditions. The combination of a particular resource status and automation flag setting may initiate an automation procedure.

Important from a maintenance and portability viewpoint, the use of the housekeeping procedures, control file, and status file allows changes to the environment-dependent names, or to certain static criteria, without requiring changes to programmed functions.

Control File

The control file contains the variable information that is usually modified when an IMS environment is defined. It contains entries that define the IMS commands that may be generated by the automation application to initiate IMS functions. For this variable information, the installation must specify the appropriate entries. When defining an XRF complex, the installation must specify both the XRF active and alternate IMS subsystems.

A sample control file is supplied with SA OS/390. IMS Automation extends the control file to support automation of IMS. Some of the additional functions provided are:

- Startups and shutdowns according to defined *service periods*
- Startups and shutdowns by groups of IMS regions
- Monitoring and maintaining MSC communication links

Here is an example of a control file entry:

```
SERVICE IMS10AA,
     DAY=(WEEKDAY,0830-1700),
     DAY=(WEEKEND,DOWN-DOWN)
```

This is the SERVICE entry that defines service periods for the IMS subsystem known as IMS10AA. In this example, IMS10AA starts at 8:30 in the morning and shuts down at 5:00 in the evening each weekday. It remains down during the weekend.

If needed, the operator interface can be used to redefine a service period for a given day.

Status File

The SA OS/390 status file maintains a log of startup and shutdown events and service periods. Along with the control file, the status file helps determine when and how automation is to occur.

The status file also preserves IMS subsystem information when NetView is stopped and restarted, which is a required function for IMS automation. For example, preserved information is used when service periods are altered through the operator interface. If SA OS/390 requires re-initialization, which reloads control file data, IMS Automation merges the operator defined service periods with those from the control file. Operator definitions take precedence over control file entries.

Typically, the status file variables do not have to be modified by the installation because they are changed only by the automation functions as various events occur. These variables indicate specific situations related to active monitoring or recovery functions. The status file is intended to isolate these variables from the ones contained in the control file, decreasing the risk of inadvertent changes during subsystem customization definition. For example, global modules will initialize the indicators used by the operational routines to keep track of the occurrence of events. Since installation would not typically change the initialization of the indicators that drive the automation, this logic-related information has been placed in the status file to isolate it from the environment-related information that would change.

State/Action Tables

A major portion of automation involves driving actions based on *events* like error messages or operator actions. One way IMS Automation does event-driven automation is by associating a single message (an event) with a single automated response such as issuing a command, writing a message, or issuing an alert (the action). However, you may want different automated actions to occur under different conditions. For example, you may want different actions to be taken depending on whether the application is active or inactive when the event occurs. IMS Automation provides *state/action tables* for this type of automation. The state/action tables associate the event with the application's *state*, (active, stopping, down, and so on). Figure 1 shows a sample state/action table.

```
1 Event

              2 State

                   3 Entry (action/new state)
/*************************************************************************/
/* Status            Init     Down      Up         Stop       Discon  */
/* State Value        0        1         2           3           4    */
/*************************************************************************/
EVENT=DFS2160I        /2       NA        NA          /2          /2
EVENT=DFS2161I    EVIECM01/3   NA    EVIECM01/3      NA          NA
EVENT=DFS2168I    EVIECM01/2   NA    EVIECM01/2  EVIECM01/2  EVIECM01/2
EVENT=DFS2169I    EVIECM01/4   NA    EVIECM01/4      /2          /2
EVENT=DFS2142     EVIECM01/3   NA    EVIECM01/3      NA          NA
EVENT=DFS2140         /3       NA        /3          NA          NA
EVENT=DFS2236I        /3       NA        /3          NA          NA
```

Figure 1. Sample State/Action Table

The state/action table works like this:

1 Event
When an event occurs, IMS Automation finds the event in the state/action table. For example, when message DFS2161I is received, IMS Automation finds `EVENT=DFS2161I` in the state/action table.

2 State
IMS Automation checks the state of the application. For this example, the application is in its initial state; the state value for the initial state is 0.

Notice that the other valid states in this sample table are down (state value 1), up (state value 2), stopping (state value 3), and disconnected (state value 4).

3 Entry (action/new state)
IMS Automation finds the associated entry. For this example, the entry is located in the same row as `EVENT=DFS216I` and the same column as the initial state (`INIT`). The entry IMS Automation finds is `EVIECM01/3`, which gives IMS Automation two pieces of information: the action to take (EVIECM01) and the new state value (3). Thus, IMS Automation issues the routine EVIECM01 and then sets the state value for the application to 3, which is "stopping" in this example.

As the application's state changes, IMS Automation records the corresponding events that have occurred. By identifying an action to be performed with each change of state, IMS Automation compiles a complete package which:

- Maintains a history of events
- Identifies when to execute an action
- Identifies what action to execute

Program-to-Program Interface

NetView's program-to-program interface (PPI) provides the ability to communicate between a NetView application and other address spaces on the same host, such as IMS. The PPI enables NetView automation to include cooperative execution of commands in IMS and command processors in NetView.

IMS Automation uses the NetView PPI. Both NetView and IMS have IMS Automation program-to-program interface components.

AOI Exit

IMS Automation uses the IMS automation operator exit, DFSAOUE0, to trap messages destined for the IMS MTO operator. These messages can then be routed to NetView, suppressed from the MTO console, or sent to the MVS system console.

To accomplish this, IMS Automation supplies a program which is linked into the IMS nucleus. This program is supplied with an alias of DFSAOUE0, so that the IMS system gen can be run and then, by simply pointing at the right library, the IMS Automation version of DFSAOUE0 is linked into the nucleus.

Control over which messages to suppress, route, or send to the console is provided thru an assembled table. The exit program loads the table during initialization. This table, named EVISPINM, can be modified.

Many IMS systems already have an DFSAOUE0 exit. IMS Automation's exit has been written to call another exit. The IMS Automation exit will call a program named EVIAOUE0. The EVIAOUE0 exit does not have to be linked into the nucleus, only into an IMS library accessible to the control region. The name of the program that the IMS Automation exit calls can be changed by altering the EVISPINM table.

Operator Interface and Single-Point-of-Control

Users monitor and control IMS on multiple domains through the operator interface. Many of the operator tasks start pre-defined automation processes, maintaining automation policies consistently.

The operator interface is panel-driven, providing the advantages of standardized operating procedures, improved accuracy and control, reduced complexity and error potential, and improved productivity.

The primary feature provided by IMS Automation to enhance operator productivity is single-point-of-control. With single-point-of-control, IMS Automation operators can control multiple IMS regions from one NetView terminal on the focal point. Operators can start or stop any IMS subsystem, defined in the control file, within a single Central Processing Complex (CPC) or across multiple CPCs.

Through utilization of the operator interface panels, status information for the entire IMS complex is available at a glance. The operator interface panels support most master terminal operator functions and all functions of IMS Automation.

Controlling and Monitoring Multiple Domains

SA OS/390 allows you to define gateways and focal point systems that are used to:

- Forward automation messages from one domain to another
- Use SDF from a focal point system to show the status of resources from multiple domains

IMS Automation implements single-point-of-control through the use of gateways and focal points. Operators can control and monitor IMS status of multiple domains from a focal point. Refer to the SA OS/390 documentation to learn how to set up gateways and focal points.

Message Handling

IMS Automation has the following message types:

- Panel messages
- Operator notification
- Tracking

Panel messages display at the bottom of the panel. The default profile displays the messages without numbers. Through the IMS Automation support functions, you can override the default and set message ID options to display the message numbers.

Notification messages inform the operator of critical situations. SA OS/390 handles notification messages like this:

- It sends notification messages to SDF, where the subsystem status is dynamically updated and displayed on an SDF panel.
- It sends notification messages to an operator ID or IDs defined with notification operator (NTFYOP) SA OS/390 control file entries. These entries identify the operator IDs of the people who receive notification messages. The messages can optionally be held on the screen until the operator clears them.

  In a distributed environment, the notification operators can be defined at the focal point host. In a single NetView host environment, the single host is considered the focal point host.

  When a notification message is received and none of the notification operators are signed on to NetView, the message is sent to an MVS system console.

Tracking messages are informational and are recorded in the NetView log.

Critical Messages Manager

IMS Automation implements SDF through the the Critical Messages Manager facility.

From the operator interface, users can access SDF and can display systems and system resources. Color indicates the resource priority and criticality of resource status. Red usually indicates that a resource is not active when it should be, but colors can be customized. SDF is dynamically updated as messages are received or resource status changes.

When IMS Automation is installed on your system, the resource item **IMS** is added to the reserved SDF words. Figure 2 uses an **I** under the heading “Features” to show the status of certain types of IMS errors.

```
SYSTEM                AOC/MVS SUPPORT SYSTEMS

   System   Subsystems   WTORs     Gateways   Spool        MVS Comps  Features

   DALLAS                                                             I  C  O

   RALEIGH                                                               C

   CHICAGO                                                            I

   DENVER                                                                   O

   PHOENIX  FPIMSEA      IMS401E   AOF05I     JES                     I  C  O
```

Figure 2. SDF Sample Panel

If the **I** displays red, a critical error has occurred in at least one of the following categories:

- IMS Critical Message
- IMS Archive
- IMS MSC
- IMS OLDS
- IMS RECONS
- IMS Timers
- IMS Transactions
- | IMS CF Structures

For information on setting up the Critical Messages Manager, see “Setting Up SDF for IMS Automation” on page 25. For information on using the Critical Messages Manager, refer to *AOC/MVS IMS Automation Feature Operator’s Guide*.

Operator Interface Panels

The following section describes the IMS Automation operator interface panels. The *AOC/MVS IMS Automation Feature Operator's Guide* contains detailed descriptions of these panels.

Main Menu
This menu is the entry point for all IMS Automation functions. It may be accessed from the NetView NCCF panel by entering **IMS** on the command line.

Inquire Subsystem Components
Selecting an option on this panel will display one of the following panels containing information for a specified subsystem:

Subsystem Information
This panel provides detailed information for the selected subsystem including:

- Subsystem status since a time and date
- Job name
- NetView domain
- VTAM information
- DC status
- Last and next start
- Last and next shutdown

Subsystem/Defined Regions Display
This panel provides selected subsystem and dependent regions information, including subsystem status, subsystem type, and job name.

Subsystem/Active Regions Display
The results of the /DIS ACTIVE command are displayed for the selected subsystem's dependent regions, including subsystem status, subsystem type, job name, and task name.

Shutdown Status Display
The shutdown status of the selected subsystem displays with a list of all active terminals.

Takeover Reason Code
This panel lists the code and causes for an XRF takeover.

Start Functions
Displays a selection list of start types and start options. Start type and options may be specified, or defaults may be used. From this panel, operators initiate an automated or a user-defined start command. Each automated start type is defined in the control file.

When an operator selects the manual start option, a pop-up box opens where the operator can either enter a specific IMS restart command or select from a list of user-defined start commands. User-defined startups are defined in the control file with the USERSTART entry.

Operator Shutdown
This panel provides optional selection of the shutdown type, control, and option, or defaults may be used. Automated shutdown is initiated using this panel.

After a shutdown type is initiated, IMS Automation returns the Shutdown Confirmation panel. The operator may confirm or cancel the selected shutdown.

Triggers

IMS Automation allows system programmers to define required conditions for startup and shutdown. IMS Automation checks these trigger conditions each time a startup or shutdown is initialized. If the conditions are not met, IMS Automation alerts the operator and asks whether to proceed or to cancel the requested action.

Service Periods

Regular start and stop times are defined for each subsystem in the IMS Automation control file. Operators can display one- and seven-day service period schedules for specific IMS subsystems. By typing over the service period timeframes, operators can force a temporary override of scheduled startups and shutdowns.

Master Terminal Operations

Through the Master Terminal Operations, IMS Automation provides access to DBCTL and TCO functions for users of IMS 3.1 or higher.

When you select DBCTL and specify a subsystem, IMS Automation lists the ten most recently issued DBCTL commands for that subsystem. Typing S before one of the commands selects it; pressing ENTER displays the reply issued to that command.

Selecting option 2 from the Master Terminal Operations panel displays the TCO Main Menu panel. From this panel you can:

- Load a TCO member
- Enable/Disable the TCO facility for a subsystem
- View TCO control file entries for a subsystem

The Master Terminal Operations panel also enables operators to display a selected subsystem's message regions. Displayed information includes request, subsystem status, and job name. Operators can start, stop, and maintain IMS message regions from this panel.

Critical Message Manager

The Critical Message Manager is IMS Automation's implementation of SDF. When a message is issued that has been user-defined in the NetView automation table as critical, the message content is saved and accumulated for future interrogation. When an operator selects the Critical Messages option on the Main Menu, an SDF panel lists message categories. If critical messages have accumulated for a specific category, the category will be highlighted. Operators can tab to the category and press PF2 to display detail of the critical messages in scrollable format.

Broadcast

The Broadcast panel allows operators to enter a message and specify destination. This function is useful to alert system users of changes in operation schedule or unplanned outages.

Support Functions

With this option, users can:

- Set a trace for their operator ID or for the domain
- Activate the program-to-program interface
- Set messages to display with their message IDs
- Set the interface panels display options

Local Functions

IMS Automation provides your system programmer the option to integrate locally developed functions into the IMS Automation interface. If local functions have been implemented, you would select this option to display a menu of them.

Processing Environment

This section describes automation functions applicable to XRF IMS configurations. Automation functions specific to the XRF IMS environment include a two-CPC approach and surveillance/takeover capability.

Two-CPC Approach: The XRF IMS automation implementation goes beyond the intra-CPC, message-driven, passive approach to automation and focuses on a two-CPC approach, using active monitoring logic to communicate status information between the two NetViews in an XRF environment. This exchange of information between the two CPCs enables NetView-based CLISTs and REXX procedures to complement the extensive automation functions of XRF during takeover. Furthermore, a key function included in the automated application is a flexible scheme for automatically synchronizing IMS regions between the *active* and the *alternate*, for startup, shutdown and takeover operations.

Surveillance/Takeover Capability: Facilities are provided in an XRF environment to forcibly terminate a failing *active* XRF partner, automatically respond to the IMS “I/O Prevention Completed” message, cancel DBRC on the old *active*, and automatically start the *alternate* (new *active*) XRF partner after takeover. IMS dependent message region synchronization is maintained between an active IMS subsystem and an alternate IMS subsystem in an XRF environment. IMS subsystem failures are detected. The IMS control region, DBRC address space, DL/I address space, and IRLM address space are automatically recovered.

Another key point is that the automation application supports a two-CPC implementation, using a NetView-to-NetView task (NNT) session to communicate between the *active* CPC and the *alternate* CPC.

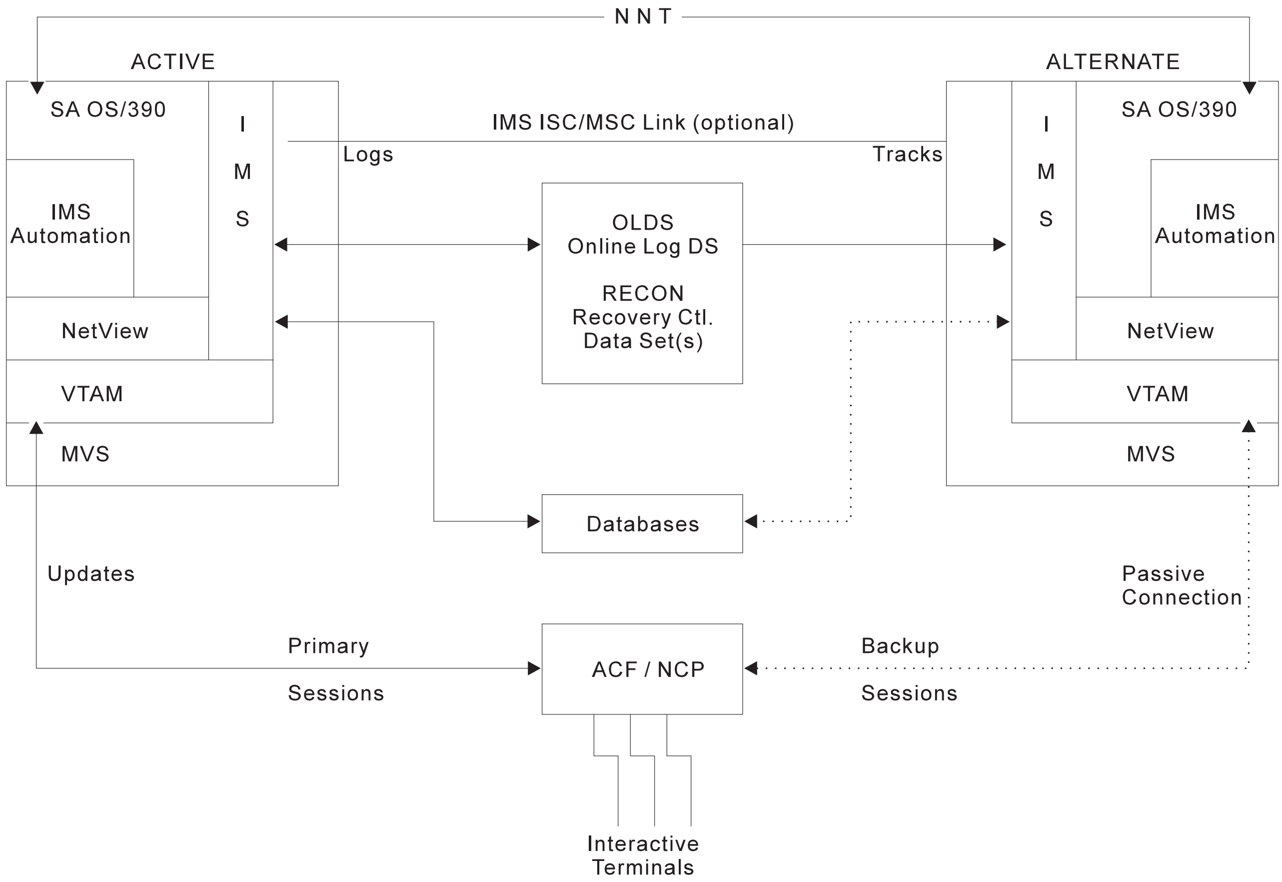

Figure 3. IMS Automation in a NetView environment

Functional Overview

The following sections provide an overview of the startup, shutdown, recovery, and takeover functions of IMS Automation.

Startup Overview

By executing a single NetView CLIST, an IMS Automation operator performs the numerous functions associated with IMS startup. The operator can start both non-XRF and XRF active and alternate IMS subsystems through the Start Functions panel, shown in Figure 4 on page 15. (See “Processing Environment” on page 13.)

```
EVIKI100            IMS Automation: Start Functions          Page:   1 of   1
                                                             Date:  05/22/95
Subsystem, Group or Domain   IMS05Z__ (? for list)           Time:  17:08:00

Select a start type  . . .   _ 1  Default start type
                               2  Auto start
                               3  Cold start
                               4  Build the messages queues
                               5  Load MSDB during warm start
                               6  Supply the IMS restart command

******************************************************************************
* Warning: The following options are used for XRF systems only.             *
******************************************************************************

Select a start option  . .   _ 1  Start XRF active only
                               2  Start XRF alternate only


Next start  : 05/29/95   11:55         Next shutdown   : 05/29/95   17:00

Command ===>
F1=Help      F2=End       F3=Return    F4=IMS Menu              F6=Roll
```

Figure 4. Start Functions Panel

A BUILDQ start implies that all messages on the queues are to be saved across restarts.

When performing an XRF IMS BUILDQ start, it is crucial that the operator start the correct procedure. This procedure must be the one associated with the XRF partner that was the *active* during the previous shutdown. If the wrong procedure is used, the integrity of the queues will probably be compromised. To prevent the operator from specifying the wrong HSBID, this process has been automated. The automation application stores the *active's* HSBID for later use at restart time. In this way the automation logic can prevent startup of the wrong control region, and thus avoid using the wrong set of queues.

Shutdown Overview

The operator can invoke the Operator Shutdown panel, shown in Figure 5 on page 16, to stop non-XRF and XRF IMS subsystems. The automation for shutting down the XRF complex focuses on allowing both the *active* and the *alternate* to be shut down in a synchronized manner.

The XRF shutdown function implemented in the sample provided with IMS Automation allows the operator to stop an active IMS subsystem and its alternate IMS subsystem. In an XRF environment, it is not possible to stop the active IMS subsystem without also stopping the alternate IMS subsystem. The alternate IMS subsystem could be stopped on its own.

During shutdown, the alternate IMS subsystem is terminated first. When the *alternate* shutdown is complete, its NetView will communicate to the *active's* NetView that the *alternate* has completed termination. Shutdown for the *active* will proceed.

```
EVIKT100                 IMS Automation: Operator Shutdown
                                                                       Date: 05/22/95
Subsystem Name . . . . . .  IMS10A1_ (? for list)                      Time: 17:11:00

Shutdown Type  . . . . . .  1 1  Normal shutdown
                              2  Immediate shutdown
                              3  Force shutdown
                              4  Cancel pending shutdown request

Restart subsystem  . . . .  2 1  Yes
                              2  No
                              3  Control

Broadcast message  . . . .  2 1  Yes
                              2  No


Press F8 for shutdown option, current value:  *

Next start  : none                    Next shutdown   : none

COMMAND ===>
F1=Help     F2=End      F3=Return                              F6=Roll
            F8=Options
```

Figure 5. Operator Shutdown Panel

Recovery Overview

Recovery from an IMS control region failure is driven by various indicators defined in the control file and status file. Errors detected during startup and shutdown processes are analyzed, resulting in automated recovery actions or the issue of error notification messages to the operator. Other recovery conditions addressed by IMS Automation include:

- Transaction/program abends

  Automation has been developed to restart the transaction/program when message DFS554A is received, indicating that an abend has occurred. Only those transactions or programs specified for recovery in the control file will be restarted.

- Archiving

  IMS Automation monitors for timely completion of archive jobstreams and notifies operators when the jobs do not complete.

- OLDS failures

  IMS Automation notifies operators when OLDS have errors and are unusable. IMS Automation also monitors to make sure that an adequate supply of OLDS are available and starts spare OLDS when needed.

- RECON errors

  IMS Automation monitors to make sure that a spare RECON is available and notifies operators whenever no spare is available. IMS Automation also notifies operators whenever there is a failure with RECON dual-copy mode.

- MSC link failure

  MSC link recovery depends on several messages and the definition of the MSC control statements in the control file. Wherever possible, the failing link is restarted.

Recovery is designed to function in either a non-XRF or XRF environment. In an XRF environment (refer to “Processing Environment” on page 13), if the active IMS subsystem fails and an alternate IMS subsystem has been started, the recovery will be handled by XRF, with the automation application supplementing the XRF function. If an *alternate* has not been started when an IMS failure occurs on the *active*, the *active* will be restarted by the automation application for both the IMS control region and the dependent regions.

If the automation application does not know whether an *alternate* started, it will send a message to the operator. The situation of not knowing about the *alternate* will occur only if the NetView-to-NetView communication link has been lost. The message will inform the operator about the loss of NetView communication. The message will also indicate, if an *alternate* started, whether XRF will handle the recovery automatically or the operator should manually recover IMS.

Automatic Restart Manager (ARM)

Version 1 Release 4 of SA OS/390 provides support for MVS/ESA SP 5.2 automatic restart manager (ARM), which allows a subsystem to be restarted on another system image of the sysplex. Underlying SA OS/390’s support of ARM is the concept of association that defines the primary image on which the subsystem is expected to run, and the secondary (or fallback) images on which the subsystem can run if the primary system is lost. For more information on this, see “How SA OS/390 Coordinates with MVS Automatic Restart Manager” in *System Automation for OS/390 Customization*. It is recommended that the SA OS/390 definition dialogs be used when specifying the system association and ARM element name. In IMS Automation, a subsystem name can occur in each ACF file for more than one system, but it must have a primary association with **only** one system.

When system associations have been defined for a subsystem managed by IMS, the subsystem is initially brought up on the primary system, and IMS’s automation continues to manage it there until one of the following occurs:

- ARM moves the subsystem to a secondary system because the primary system becomes inoperable
- The operator changes the automation status to MOVED

Following an ARM move to a secondary system, the secondary system takes control of the subsystem for the remainder of the current service period as defined on the secondary system. When the service period ends or when a SHUTSYS command is issued, control passes back to the primary system (whether the primary system is available or not). If the primary system is re-IPLed while the subsystem is still active on the secondary, the primary does not restart the subsystem.

It is recommended that each system use the same service periods and triggers. You can accomplish this for your policy data by using the same %INCLUDE statements in the control file for each system.

Takeover Overview

ARM and XRF can be used alone or together to provide system recovery. ARM provides services to support an XRF environment by preventing conflicting actions by ARM and XRF. XRF provides high availability for its users by providing an environment where an active production system has an alternate system that is ready to take over processing in the event of the active failing. If the alternate detects a failure of the active, then it initiates an XRF takeover.

IMS registers both the active and alternate XRF systems with ARM, but it only issues an associate for the alternate XRF system. For subsystem failures ARM restarts the alternate XRF system if it fails. It does not attempt, however, to restart the active XRF system if it fails. If the active XRF system fails, then the alternate XRF system takes over for it. Therefore, an ARM restart is not necessary.

XRF Takeover

A takeover condition is an event that causes an alternate IMS subsystem to request a takeover. In an installation that runs XRF, an IMS control region abend is always a takeover condition. When tailoring the IMS subsystem, the system programmer specifies whether any of the following failures are also takeover conditions:

- An MVS failure, loop, or wait state
- A CPC failure
- A VTAM failure
- IRLM failure

Not all failures at an IMS installation can be takeover conditions. XRF does not address the outages caused by failures of service elements you do not duplicate. For instance, XRF does not respond to:

- A channel or link failure that causes a break in communication between the CPC and the communication controllers or DASD
- Failures in the telecommunication network, such as communication controllers, NCPs, lines, and terminals
- Inter-system failures, such as those caused by JES3 or CTCs
- Loss of or damage to the IMS databases
- A power failure that affects both CPCs in the XRF complex
- Failures of user catalogs that point to data sets, such as databases

One of the primary goals of IMS Automation is to eliminate, where possible, any human intervention during the takeover process for an XRF environment. Although XRF provides a very comprehensive approach to handling the workload transfer, a few exception conditions require operator interaction, and operational errors could cause severe problems. IMS Automation was designed to automate these actions and to eliminate the human element where possible.

With the subject of an XRF takeover, we are by definition talking about an “extra-system” automation environment and the need for communication between two CPCs. In the case of a takeover, the NetView-to-NetView session being used to pass information back and forth between the *active* and the *alternate* CPCs was judged to be inadequate, as it relies on having VTAM active. Because VTAM failure is one of the failure types that the redundant-resource design of XRF supports, IMS

Automation was designed to provide support for a takeover without VTAM being operational.

Although the NetView-to-NetView session is used to check the dependent regions on a continuous basis, it is not used at all during the takeover.

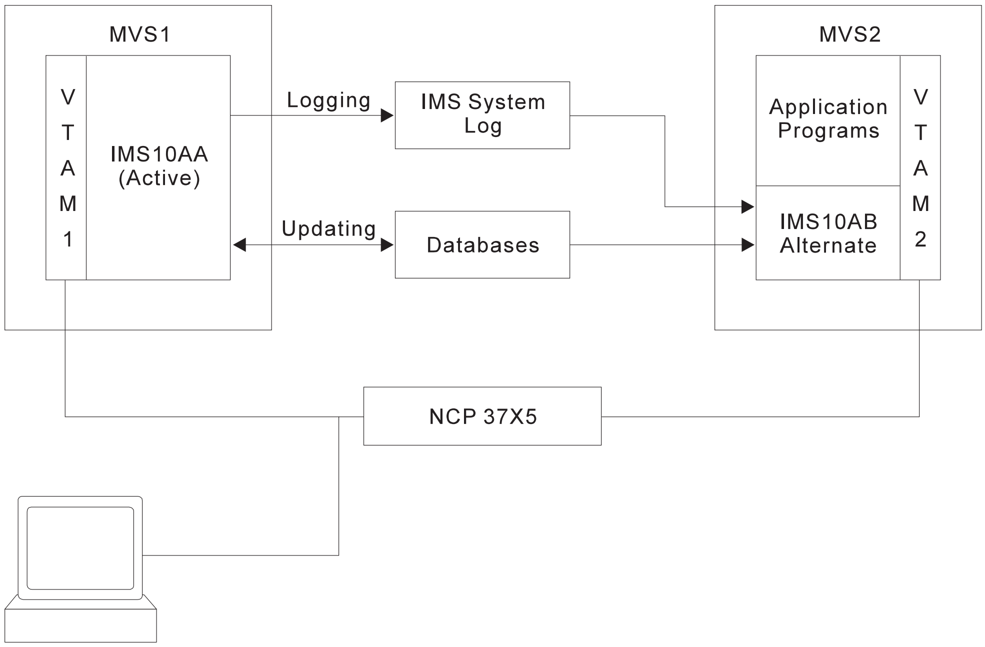

Figure 6. XRF Complex Before Takeover

In Figure 6, and Figure 7 and Figure 8 on page 20, the XRF complex is shown before, during, and after takeover. Illustrated takeover activities are discussed below:

1. IMS10AA processes the high priority work that comes from the remote terminals. It updates the databases and also records its activity on the IMS system log.
2. IMS10AB tracks the *active* by monitoring the records on the IMS system log. It also opens backup sessions for Class 1 terminal users who log on to the *active*. To maintain an environment identical to that in the *active*, IMS10AB updates many control blocks and message queues in the *alternate* to reflect those in the *active*. Processor capacity and storage not used by this activity support the application programs.
3. When IMS10AA abends, the takeover begins. Depending on XRF's demand for real storage, MVS2 might swap out the application programs. IMS10AB shifts the production workload to itself and begins to serve Class 1 and Class 2 terminals. Problem determination activities can begin on the failing IMS10AA.
4. While IMS10AB recovers data and tells NCP to switch sessions on Class 1 terminals, IMS10AB and MVS1 prevent IMS10AA from writing to the IMS system log and the databases. IMS10AB isolates the log and proceeds with the takeover. At the same time, MVS1 performs I/O prevention, it ensures that all new I/O requests to the databases from IMS10AA return without being executed. When MVS has completed or cancelled all existing I/O requests to the databases, it notifies the operator.

5. The takeover is complete when all the users at Class 1 terminals can communicate with IMS10AB and can enter transactions and receive replies from their IMS applications. When IMS10AB learns that the failing IMS10AA cannot write to the databases, it stops protecting them.

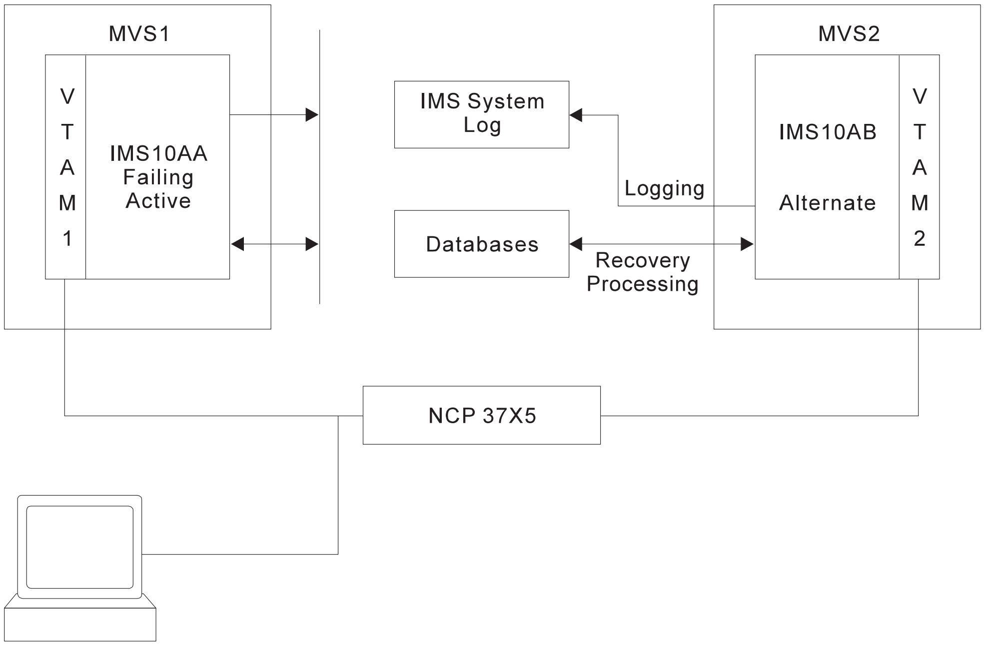

Figure 7. XRF Complex During Takeover

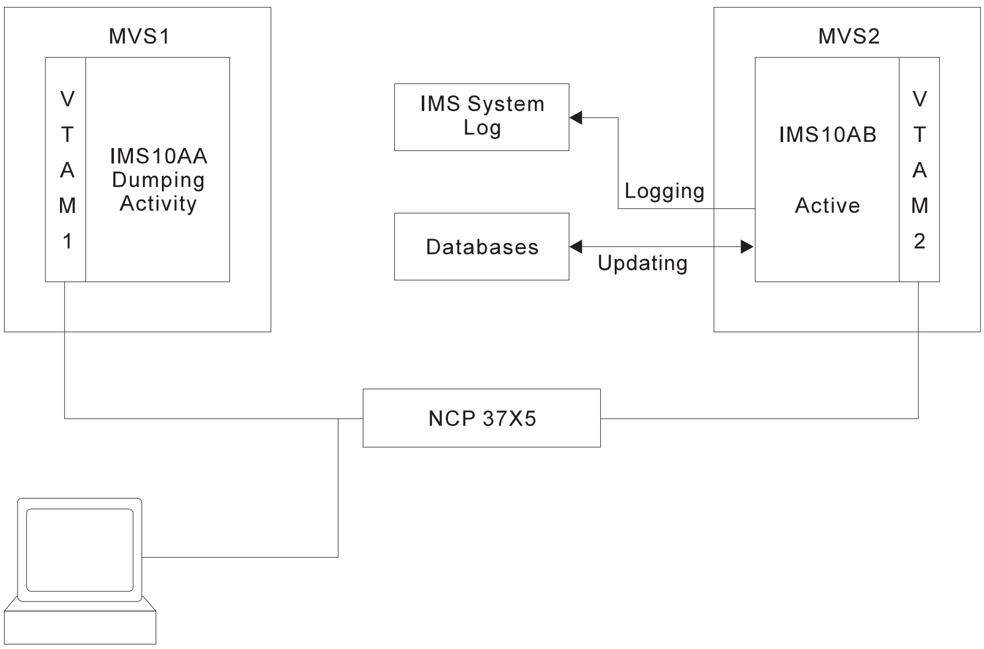

Figure 8. XRF Complex After Takeover

XRF and ARM Takeover

The following cases tell how XRF and ARM takeovers work.

XRF Active Fails While XRF Alternate Is Available: When an XRF pair are both running and the active subsystem (IMS) fails, an XRF takeover occurs. ARM does not attempt to restart the failing active subsystem.

ARM Recovers from Failure of XRF Alternate Subsystem: When an XRF alternate subsystem fails, ARM restarts that subsystem on the same system.

Entire System Fails: When an XRF pair are both running on a single system and the entire MVS system system fails (both the IMS *active* and *alternate* fail), then ARM restarts an IMS on another system identified by the ARM element group. This IMS must be identified by the ARM element group defined as an XRF partner associated with the IMS that failed.

ARM Recovery of Failure of XRF Alternate When Entire System Fails: When the system where the XRF alternate is running fails, ARM restarts an IMS on another system identified by the ARM element group. This IMS must be identified by the ARM element group defined as an XRF partner associated with the IMS that failed.

FDR Environment

This section describes automation functions applicable to FDR enabled IMS environments.

The automation functions provide cross-system support for IMS and FDR startup, and cross-system automation of the messages associated with IMS failure and subsequent FDR recovery. This is achieved by invoking System Automation for OS/390 V1R3 sysplex services.

Recovery Capability: FDR IMS Automation implementation will support automatic response to the IMS DFS4167A WTOR message when the IMS I/O prevention completed message AVM006I is received during recovery processing.

Functional Overview

The following sections provide an overview of the FDR region startup, shutdown and recovery functions provided by IMS Automation.

Startup Overview

During IMS startup, when the control region has completed initialization, the IMS subsystem will be put to "UP" status. Message DFS4190I is trapped to indicate if FDR is enabled. If IMS is FDR enabled, automation will initiate a startup of the FDR region in the NetView domain as defined in the Automation Control File.

Recovery Overview

IMS Feature provides an automated response to allow FDR recovery processing to proceed. Assume that IMS is running on system SYS1 and FDR on system SYS2. The sequence of events is as follows:

1. When IMS on system SYS1 abends, WTOR message DFS4167A is issued by the FDR region on SYS2 indicating that it is waiting for the failing IMS region to complete I/O Prevention.
2. When the failing IMS on system SYS1 completes I/O Prevention, message AVM006E is issued by the Availability Manager.
3. Automation then replies 'UNLOCK' to the WTOR message DSF4167A, to allow FDR on SYS2 to commence recovery processing.

Shutdown Overview

Automation then replies 'UNLOCK' to the WTOR message DSF4167A, to allow FDR on SYS2 to commence recovery processing.

Chapter 2. How to Implement and Use IMS Automation

In an automated environment, startup, shutdown, and recovery procedures are initiated by events, such as messages, operator requests, and timers. IMS Automation provides standard procedures that respond to these events. However, you can customize these procedures for your specific needs.

To help you get started, the following samples or source are provided. The installation procedures and how-to descriptions explain when to use them.

Table 4. Samples and Source Used for Customization

Member	Description
EVIS2CFG	The %INCLUDE member that contains definitions for IMS automation. It is merged into the SA OS/390 control file.
EVIDBCTL	Set of definitions for the DB control region.
EVITREE	The IMS Automation panel tree definition. See "Setting Up SDF for IMS Automation" on page 25 for a description of this member.
SDF panels	Several SDF panels are included in the IMS Automation package. See "Setting Up SDF for IMS Automation" on page 25 for a description of these panels.
EVIOPF	IMS automation specifications that are merged into DSIOPF.
EVICMD	IMS automation specifications that are merged into NetView's Command Model Statements member.
EVISECUR	Security definitions.
EVIDMN	IMS automation specifications that are merged into DSIDMN.
EVIPRFAO	IMS automation operator profiles that are added to the NetView profile library.
EVINTASK	Program-to-program interface definition on the NetView side of the interface.
EVISPINM	Program-to-program interface definition on the IMS side of the interface.
EVISI001	IMS PCB gen entries.
EVISI002	IMS gen required APPLCTN and TRANSACT entries.
EVISI003	IMS security gen entries.
EVISS002	State/Action table entries for MSC area.
EVISS003	State/Action table entries for OLDS area.
EVISS005	State/Action table entries for Transaction/Program area.
EVIS2IMS	Control file entries for IMS Subsystem.
EVISTSM	DSTINIT example for XRF in a dual-CEC environment.
IMSMSG00 (EVIMSG00)	Message table (%INCLUDEs EVIMEVI0 and EVIMCON0).
IMSMSG01 (EVIMSG01)	Message table (%INCLUDEs EVIMEVI1, EVIMAVM1, EVIMDFS1, EVIMDSP1, EVIMIOS1, EVIMDXR1, EVIMCON1, and EVIMCQS1).

Using Single-Point-of-Control

To understand how IMS Automation works, your understanding of gateways and focal point services is essential.

Automation networks can be configured to allow messages to be routed between systems. Focal point services, a feature of SA OS/390, is used to define the domains to which messages are sent. This allows the operator to monitor systems in a distributed environment. See Figure 9.

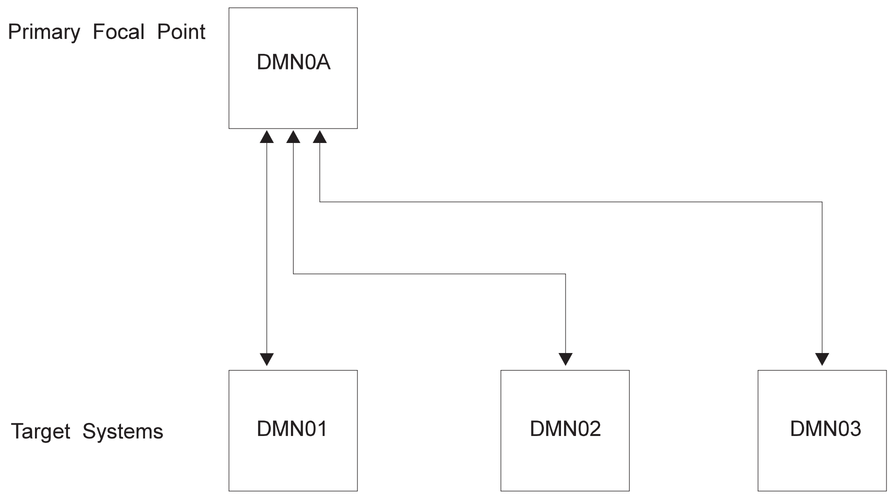

Figure 9. Focal Point Definitions in a Distributed Environment

In this figure, domain DMN0A is designated as the focal point system for domains DMN01, DMN02, and DMN03, known as the target domains. SA OS/390 uses focal point services to update the status of resources shown on the SDF panels. In the scenario depicted in Figure 9, the status of resources from domains DMN0A, DMN01, DMN02, and DMN03 would all be available on DMN0A.

Added Capabilities

Single-point-of-control adds the ability to control and to monitor all of the systems from a single operator session on the focal point system. All functions available on the focal point system IMS Automation operator interface are also available to the target systems, including inquiries, startups, and shutdowns.

How Single-Point-of-Control Works

When a target domain is initialized and a focal point system is designated for that target system, static information about the IMS subsystems, such as job names and application IDs, is forwarded to the focal point system. Dynamic information is either forwarded or retrieved when needed.

Because the focal point system is aware of the domains on which the target IMS subsystems reside, commands can be issued to those subsystems.

In essence, setting up a focal point configuration according to the descriptions in the SA OS/390 base documentation allows you to use single-point-of-control. No further definitions are required in IMS Automation.

If you are using the IMSPOST command processor to post triggers across domains, as described in “Defining IMS Automation Startups and Shutdowns” on page 30, gateways must be defined between the “from-to” domains.

If you are using the IMSPOST command processor to post triggers across domains, as described in “Defining Sartup and Shutdown Triggers” on “Defining Startup and Shutdown Triggers” on page 32 gateways must be defined between the “from-to” domains.

SPOC Requirements

Single-point-of-control requires that the subsystem names used for IMS regions must be unique across the enterprise. IMS Automation builds a table which matches subsystem names with domain names in order to route commands to the correct NetView domain. A subsystem name must match only one NetView domain. If a subsystem name matches more than one NetView domain, the results are unpredictable.

When you are using system association support, a subsystem can be defined to multiple systems, but only one system can have a primary association to it. The other associations must be secondary.

Setting Up SDF for IMS Automation

SDF uses color to represent the various subsystem resource statuses, such as error, warning, action, or informational states. Typically, a subsystem shown in green on an SDF status panel indicates that it is up, whereas red indicates a stopped or problem state.

The SDF status display panels can be tailored to present the status of system components in a hierarchical manner. The hierarchical display of status information is implemented using tree structures. A tree structure always starts with the system name as the root component and has a level number of one. The “leaves” of the tree are the monitored resources with the level numbers reflecting their dependency on each other.

Color can be propagated up or down the leaves of the tree structure based on the order of dependencies. The effect of propagation is to consolidate at the root component the status of all the monitored resources in that system. In this way, the color of the root component reflects the most important or critical status in a computer operations center. If all the monitored resources are green, the root component (the system) will be green. Refer to the following examples:

```
DALLAS DOMAIN

1  SY1
   2  SYSTEM
      3  JES
      3  TSO
      3  RMF
      3  VTAM
         4  IMS1
         4  IMS2
   2  IMS
      3  IMSM SG
      3  IMSARCH
      3  IMSMSCL
      3  IMSOLDS
      3  IMSRECN
      3  IMSTRAN
      3  IMSTIMR
      3  IMSSTRCT
```

```
RALEIGH DOMAIN

1  SY2
   2  SYSTEM
      3  JES
      3  TSO
      3  RMF
      3  VTAM
```

```
CHICAGO DOMAIN

1  SY3
   2  SYSTEM
      3  JES
      3  TSO
      3  RMF
      3  VTAM
         4  IMS5
         4  IMS6
   2  IMS
      3  IMSM SG
      3  IMSARCH
      3  IMSMSCL
      3  IMSOLDS
      3  IMSRECN
      3  IMSTRAN
      3  IMSTIMR
      3  IMSSTRCT
```

Figure 10. Tree Structures of a Distributed Environment

The three tree structures in this example represent a distributed environment. The files containing the tree structure are located on the actual system that they represent. The systems (SY1, SY2, and SY3) are at the highest level (1) in the tree structure. The word SYSTEM (level 2) is not used by SDF to represent a subsystem. It is a reserved word used to indicate that the resources under this entry are subsystems. IMS, also a reserved word, represents a group of errors that can be generated from IMS automation. The words listed under the IMS word are the reserved words used to define the group of IMS errors. Comparing these words with the panel in Figure 12 on page 28, you will see the relationship between them and the actual data displayed.

Another reserved word, SUBSYS, can be used in place of subsystem entries to generically represent all subsystems defined to SA OS/390 on that system. This is very helpful when you have many subsystems. For example, SY1 could be defined as shown:

```
DALLAS DOMAIN

1  SY1
   2  SYSTEM
   2  SUBSYS
   2  IMS
      3  IMSMSG
      3  IMSARCH
      3  IMSMSCL
      3  IMSOLDS
      3  IMSRECN
      3  IMSTRAN
      3  IMSTIMR
      3  IMSSTRCT
```

Several control file entries are used with SDF:

SDF *status*

SDF defines the priority, highlight level, and color definitions for each status. Several keywords were added to this entry to support IMS automation. See "Coding SDF—A Basic Overview" on page 29.

IMS CRITMSGS,CODE=...

An IMS Automation SDF panel lists critical IMS messages. The CODE= control file entry supports this panel by allowing you to specify which messages should be saved.

IMS Automation SDF Panels

You can use the panel definition keywords to display the distributed environment shown in Figure 10 on page 26 as shown:

```
SYSTEM                AOC/MVS SUPPORT SYSTEMS

   System   Subsystems   WTORs     Gateways   Spool        MVS Comps  Features

   DALLAS                                                             I  C  O

   RALEIGH                                                               C

   CHICAGO                                                            I

   DENVER                                                                   O

   PHOENIX  FPIMSEA      IMS401E   AOF05I     JES                     I  C  O
```

Figure 11. Data Center Systems Panel—Sample

Figure 11 could be your primary panel that lists the systems and their status. The color of DALLAS, RALEIGH, CHICAGO, DENVER, or PHOENIX will reflect the most critical status of any resource in that system. DALLAS, CHICAGO, and PHOENIX, for example, have the letter **I** under the features column. (The RALEIGH and DENVER systems do not have IMS.) This letter represents the "2 IMS" entry in the tree structures. (Do not confuse this with IMS subsystem entries, which are listed under "2 SYSTEM").

Some of the panel definitions for this panel would look like this:

The item identifier, in this case, the system name as defined in ENVIRON SETUP SDFROOT=. This is known as the root component.

The row that this item displays on.

The starting column for this field.

The ending column for this field.

The 'DOWN' panel name to display with PFB.

```
SF(SY1.SYSTEM,05,05,10,N, ,SYSSY1)
ST(DALLAS)
```

The actual data displayed on the panel.

The item identifier. Because this is not a root component, it is a combination of the root component and this item.

IMS Automation supplied definitions for the IMS Monitor panel, to be selected by positioning the cursor at this entry and pressing the 'DOWN' PF key, PF8.

```
SF(SY1.IMS,05,72,73,N, ,EVID0001)
ST(I)
```

The actual data displayed on the panel.

If you select the letter `I` on the Data Center Systems panel, the following panel displays (assuming you are using the default sample definitions):

```
EVID0001               IMS MONITOR PANEL


        Critical Messages

        Archive

        MSC Links

        OLDS

        Recons

        Timers

        Transactions

        CF Structures



                                                                    12/06/95 13:04
 ===>
PF1=HELP 2=DETAIL 3=END 4=HWD  6=ROLL 7=UP 8=DN                             12=TOP
```

Figure 12. IMS Monitor Panel

This shows several categories in which IMS status is important. If the letter I shown on the Data Center Systems panel were red, then at least one of the items on the IMS Monitor panel would also be red, indicating that important messages had been logged in that category. To display these messages, tab to the red item and press PF8. If the full message does not display, press PF11 to shift to the right. To view the detail of a message, tab to the message and press PF2. This displays a panel such as the following:

```
                 ---- DETAIL STATUS DISPLAY ----
                                                     1 OF    6


      COMPONENT: DFS3257I             SYSTEM   : SY1

      COLOR    : GREEN                PRIORITY :      503

      DATE     : 12/14/95             TIME     : 16:13:39

      REPORTER : AUTIMS               NODE     : CNM01

      REFERENCE VALUE: IMS10AA-DFS3257I

       JOB IMS10AA - DFS3257I ONLINE LOG NOW OPENED ON DFSOLP00 IMSA







 ===>
1=HELP    3=RETURN        6=ROLL 7=UP 8=DOWN 9=ASSIST 10=DELETE 11=BOTTOM 12=TOP
```

Figure 13. Detail Display of Critical Message

If a panel displays `1 of x` in the upper right corner of the screen, where x is a number more than 1, then use PF8 to scroll forward and view information on subsequent panels. Press PF7 to scroll back.

Coding SDF—A Basic Overview

The color definitions indicate which color to use on the IMS Monitor panel when a message is logged against a specific category. The only category in which priority has any meaning for the IMS Monitor panel is IMS Critical Message. The default control file specifications assign messages ending in I as having the lowest priority. Messages ending in W have the next highest, messages ending in E have the next highest, and messages ending in A have the highest priority. The color displayed on the IMS Monitor panel IMS Critical Message item depends on the message logged with the highest priority. So, if a message ending in A is logged, the IMS Critical Message item will turn to the color defined for those messages (probably red), overriding any other message color.

The following keywords were added to the SDF control file entry to define priority and colors for the items displayed on the IMS Monitor panel:

IMSTRAN IMS transactions
IMSTIMR IMS timers
IMSARCH IMS archive problems
IMSMSCL IMS MSC link recovery

IMSOLDS	IMS OLDS problems
IMSRECN	IMS RECON problems
IMSSTRCT	IMS CF structures problems
CRITMSG	The default critical messages definition (IMS Critical Message)
CRITMSGA	Messages ending in A (IMS Critical Message)
CRITMSGE	Messages ending in E (IMS Critical Message)
CRITMSGW	Messages ending in W (IMS Critical Message)
CRITMSGI	Messages ending in I (IMS Critical Message)

Another SA OS/390 control file entry that was added to support IMS automation is the CODE entry. This entry allows you to specify which messages you want displayed on the panels that list messages by category, thus allowing you to filter out some of the messages.

Defining IMS Automation Startups and Shutdowns

When you define IMS Automation startups and shutdowns, consider the following concepts and how you may wish to use them at your facility.

Grouping IMS

The operator interface allows the operator to invoke startup and shutdown processing by subsystem, subsystem groups, or domains. A subsystem is identified by the symbolic name as it is known to SA OS/390 (for example, IMS10AA). A domain is known by the NetView domain name. A subsystem group name is defined with the control file entry IMSGROUP.

```
IMSGROUP IMSGRP1,MEMBER=IMS10AA,MEMBER=IMS10AB,MEMBER=IMS10Z
```

In this example, IMS10AA, IMS10AB, and IMS10Z are the subsystem names. IMSGRP1 is the group name for these subsystems. When single-point-of-control is used, these subsystems can be distributed across domains. The startup or shutdown initiation process starts with the first subsystem listed, then cycles through each subsystem in the list. Service periods, triggers, and XRF relationships are checked for each subsystem before the process is initiated. The operator has the option of overriding some of these checks.

Parent/Child Relationships in a Startup or Shutdown

Because dependent relationships exist among automated resources, SA OS/390 allows you to establish parent/child relationships between resources. These relationships control the startup and shutdown sequence for the affected subsystem resources.

During NetView initialization, the highest level subsystem resources in the hierarchy are the first selected for startup processing. These resources are identified by those subsystems which either have PARENT=NONE coded or which do not have a PARENT defined in the SUBSYSTEM control file entry.

The next resources to be started are those resources whose parents have completed startup. Refer to the following example:

```
SUBSYSTEM IMS10AB,JOB=IMS10AB,PARENT=VTAM...
```

This entry states that IMS10AB cannot be started until VTAM is active. Once VTAM is active, IMS Automation will allow a startup for IMS10AB **only if** it is within a service period window and if any defined trigger conditions are met. Otherwise, it will wait until the service period window is open to invoke the startup.

If the IMS Automation operator shutdown panels are used to issue the shutdown, service periods and triggers are considered. If SA OS/390 or the operator uses SHUTSYS to start a shutdown process outside IMS Automation and an IMS subsystem is within the scope of the shutdown request, then service periods are not taken into consideration.

IMS Automation supports SA OS/390 parent/child relationships as provided with the SA OS/390 product with the following restrictions:

- An IMS control region in a DB/DC environment must have PARENT=VTAM defined in its SUBSYSTEM control file entry, or must have a parent or a grandparent with PARENT=VTAM. For a DB control region, however, since it does not use VTAM, PARENT can be set to something else, for example, JES2.
- Certain IMS dependent regions (DBRC and DLISAS), must be defined as direct dependent children of the control region and must have STARTOPTIONS=PARENT and SHUTOPTIONS=PARENT defined.
- Any region defined with STARTOPTIONS=PARENT or SHUTOPTIONS=PARENT must be defined as a direct dependent of the control region and must not have multiple parents defined.
- A dependent region that is defined as a dependent of more than one control region must have the ENVIRON CTLNAME parameter coded to enable IMS Automation to determine which control region owns the dependent region.
- IMS Automation does not support one-shot, transient subsystems as dependent regions. Any dependent region with a status of ENDED will be reset to a status of DOWN when the control region starts. The only IMS Automation subsystem type that supports one-shot, transient subsystems is SUBTYPE=OTHER.
- CQS and FDR regions cannot be defined as dependent regions.

Defining the Service Periods

Service periods are the intervals of time between the startups and shutdowns of a given IMS subsystem. They are defined with control file entries, as shown:

```
SERVICE IMS10AA,DAY=(WEEKDAY,08:30-17:00)
```

which states that IMS10AA is up from 8:30 to 17:00, Monday through Friday.

Service periods utilize the NetView timer interface capabilities to initiate service period processing whenever a service period (startup or shutdown) is reached. These timers are set from the control file when SA OS/390 is initialized, but they can be changed for given days through the operator interface.

Startups and shutdowns can be invoked from the operator interface. However, the IMS Automation operator interface validates that the request is within the service period. If a conflicting request is made, such as a startup outside the service period, the operator is notified of the conflict and asked to confirm the IMS startup or shutdown.

Defining Startup and Shutdown Triggers

This function provides a method of correlating external events with IMS Automation controlled startup and shutdown procedures. Both startup and shutdown triggers are used. Startup triggers verify that all defined actions have completed prior to a startup. Shutdown triggers specify external events that must occur before a shutdown.

The TRIGGER control file entry defines which triggers must be set for a shutdown or startup to occur. See “TRIGGER—Startup and shutdown triggers” on page 142 for more information. Up to 28 unique trigger names can be used for each subsystem.

IMSPOST Command Processor

IMSPOST is used to post an external event to a specific subsystem. First, it determines the domain in which the subsystem resides, then it invokes the routine to update the status file entries. See “IMSPOST—Post an external event” on page 172 for details.

Defining Triggers in the Control File

The following entries are used to define triggers and trigger conditions:

EXTCOND This optional control file entry defines the trigger names and definitions. For example:

```
EXTCOND JOBA,DESC='Payroll'
EXTCOND JOBB,DESC='Accounts receivable'
```

TRIGGER This control file entry specifies trigger conditions that signal the initiation of startup and shutdown processes for specific subsystems. For example:

```
TRIGGER IMS10AA,STARTUP=(JOBA),
                SHUTDOWN=(JOBB)
```

Posting Triggers

Triggers can be posted with user-written routines, through the automation table, with control file entries, and through the operator interface.

To post a trigger from an external event, use the IMSPOST routine. This is the most common method of posting triggers.

Triggers can be posted through the automation table. For example:

```
IF message ID = 'MSG0123'
   THEN EXEC ( CMD ('IMSPOST NAME=IMS1,FUNCTION=SET,EVENT=JOBA,TYPE=STARTUP'))
```

To post triggers with control file entries, define the following in your control file:

EXTCOND This entry can be used in conjunction with the UNSET keyword to unset a trigger for a subsystem at specific points, including:

- When a startup is initiated
- When a startup is complete
- When a shutdown is complete

CMD/REPLY This control file entry sets or unsets triggers for a specific subsystem when certain messages are received by invoking the IMSPOST command processor.

To post triggers through the operator interface, use the Trigger option, listed on the IMS Automation main menu. This option allows the operator to:

- View trigger specifications for each subsystem
- View the condition of each trigger specification
- Post triggers (set or unset them)

The following example illustrates a possible TRIGGER entry in the control file, where IMS10AA is the subsystem:

```
TRIGGER IMS10AA,STARTUP=(JOBA,JOBB,JOBC),
                STARTUP=(SERVICE,JOBD),
                SHUTDOWN=(JOBE)
```

Selecting the Triggers option from the main menu invokes the Triggers List panel. A sample triggers list looks something like this:

```
_ STARTUP   JOBA JOBB JOBC
_ STARTUP   SERVICE JOBD
_ SHUTDOWN  JOBE
```

In this example, the highlighted words indicate those conditions that are met. The first STARTUP item specifies that a startup is to be initiated as soon as all three triggers (JOBA, JOBB, and JOBC) are set, whether we are in a service period window or not. JOBA and JOBB are set.

The next STARTUP item uses the SERVICE keyword. This states that a startup is to be initiated when JOBD is set only if it is within a service window. If it is not within a service period window, IMS Automation will wait until the service period window opens to initiate the startup. When a service window opens, a startup cannot be initiated unless JOBD is set. The highlighted SERVICE word indicates that we are within a service period window. If it were not highlighted, it would indicate that we were not within a service period.

The SHUTDOWN entry states that a shutdown is to be initiated when JOBE is complete. Because SERVICE is not specified, the subsystem will shut down whether a service period window is open or not.

To post a trigger from the operator interface, enter an `S` in the command field beside the startup or shutdown item. This invokes a panel that lists the triggers for that item and allows them to be set or unset.

Using Triggers in a Distributed Environment

As described in a previous section, parent/child relationships can be used to set up hierarchical startup and shutdown structures. However, the PARENT= keyword can only be used for subsystems on the same domain. To establish parent/child relationships in a distributed environment, you can use the IMS Automation startup and shutdown triggers.

For example, to start IMS10AA, IMS10AB, and IMS10Z on different domains, in the following order:

1. Start IMS10AA on DOMAINA.
2. When IMS10AA on DOMAINA is active, start up IMS10AB on DOMAINB.

3. When IMS10AB on DOMAINB is active, start up IMS10Z on DOMAINC.

To shut down IMS10AA, IMS10AB, and IMS10Z, (which are all on different domains) in the opposite order, code the control file entries as follows:

For subsystem IMS10AA on DOMAINA:

```
TRIGGER IMS10AA,SHUTDOWN=IMS10ABDN
IMS10AA UP,CMD=(,,'IMSPOST NAME=IMS10AB,FUNCTION=SET,EVENT=IMS1UP,TYPE=STARTUP')
```

For subsystem IMS10AB on DOMAINB:

```
TRIGGER IMS10AB,STARTUP=IMS1UP,SHUTDOWN=IMS10ZDN
IMS10AB DOWN,CMD=(,,'IMSPOST NAME=IMS10AA,FUNCTION=SET,EVENT=IMS10ABDN,TYPE=SHUTDOWN')
IMS10AB UP,CMD=(,,'IMSPOST NAME=IMS10Z,FUNCTION=SET,EVENT=IMS10ABUP,TYPE=STARTUP')
```

For subsystem IMS10Z on DOMAINC:

```
TRIGGER IMS10Z,STARTUP=IMS10ABUP
IMS10Z DOWN,CMD=(,,'IMSPOST NAME=IMS10AB,FUNCTION=SET,EVENT=IMS10ZDN,TYPE=SHUTDOWN')
```

Defining Transaction Recovery

Customizing transaction recovery consists of:

- Determining which application program (TP) transactions will have recovery automation
- Identifying the batch message region (BMP) transactions that will have recovery automation
- Specifying the error threshold level at which a recovery should stop
- Identifying specific abend codes for which you want recovery procedures to occur
- Specifying the recovery procedure, which usually consists of invoking a command, a routine, and/or sending notifications to an operator

Four control file entries allow flexibility in customizing and controlling transaction recovery:

RECOVERY

The RECOVERY automation flag turns automation on or off for transaction recovery. This can be done generically or by specific transactions.

Consider the following example:

```
RECOVERY IMS10AA.TRAN,AUTO=N
RECOVERY IMS10AA.TRAN.PAYR,AUTO=Y
RECOVERY IMS10AA.TRAN.DBTS,AUTO=Y
RECOVERY IMS10AA.TRAN.BLNG,AUTO=Y
```

This control file entry states that recovery automation is to occur on IMS10AA for transactions PAYR, DBTS, and BLNG only. The first entry is generic. Because it is set to AUTO=N, automation will not occur on any other transaction on IMS10AA except for the three listed below that entry.

You can use the same syntax to turn off program recovery by replacing IMS10AA.TRAN with IMS10AA.PROG:

```
RECOVERY IMS10AA.PROG,AUTO=N
RECOVERY IMS10AA.PROG.dfsprg01,AUTO=N
RECOVERY IMS10AA.PROG.payroll,AUTO=Y
RECOVERY IMS10AA.PROG.payrol2,AUTO=Y
```

Note: `RECOVERY DFS554A, AUTO=N` is another format you can use to turn off program or transaction recovery.

ABCODETRAN

Use the ABCODETRAN control file entry to specify abend codes for which recovery action will occur. You can also code this entry generically or by specific transactions. Refer to the following example:

```
ABCODETRAN IMS10AA.TRAN,
     CODE=(PART,U3033,*,INCLUDE),
     CODE=(PART,U907,*,EXCLUDE),
     CODE=(*,*,*,INCLUDE)
```

When coded this way, recovery actions will occur for PART on IMS10AA when the transaction abend code is U3033, but recovery will not occur if the transaction abend code is U907. (INCLUDE indicates that action should occur; EXCLUDE indicates that action should not occur.) The last item, CODE=(*,*,*,INCLUDE), states that recovery will occur on all other transactions, for all transaction abend codes.

Another example would be to specify the last entry as CODE=(*,U3033,*,EXCLUDE), which states that recovery actions will not occur when the transaction abend code is U3033, no matter what the transaction is.

ABCODEPROG

Use the ABCODEPROG control file entry to define actions to be taken for program-driven batch message regions (BMP). This definition utilizes the same qualities as ABCODETRAN. Recovery uses ABCODEPROG only for the program-driven BMP regions, while transaction-driven BMP regions use the ABCODETRAN entry. Refer to the following:

```
1 ABCODEPROG IMS10AA.PROG,CODE=(*,U0456,*,INCLUDE)
                              CODE=(*,SOC1,SAMPRG11,EXCLUDE)

2 ABCODEPROG IMS10Z.PROG.SAMPRG12,CODE=(*,U0456,*,INCLUDE)
```

The first entry (**1**) defines an entry for IMS10AA program-driven BMP recovery where U0456 abends will be recovered, but S0C1 abends will not be recovered. The second entry (**2**) defines recovery for subsystem IMS10Z where U0456 abends are recovered only for program-driven BMP regions running program SAMPR612.

THRESHOLD

For transaction recovery, the THRESHOLD CRIT= keyword is used to determine when to take recovery action. This can be done generically or by specific transactions. Refer to the following example:

```
THRESHOLD IMS10AA.TRAN,CRIT=(3,00:01),FREQ=(3,00:05),INFR=(3,00:30)
THRESHOLD IMS10AA.TRAN.PART,CRIT=(2,00:01),FREQ=(2,00:05),INFR=(2,00:30)
```

When coded this way, recovery will occur for all transactions except PART until three abends occur within one minute. This is the generic or default definition. For PART, recovery actions will occur until two abends occur within one minute.

Note: The THRESHOLD settings are also used for operator notification, as described in the SA OS/390 base documentation.

To recover programs instead of transactions, you use similar syntax:

```
THRESHOLD IMS10AA.PROG,...
```

DFS554A
Use the DFS554A control file entry to define the specific recovery actions:

```
IMS10AB DFS554A,
                CMD=(TRAN,,'IMSCMD &SUBSAPPL  /STA TRAN &EHKVAR1'),
                CMD=(PROG,,'IMSCMD &SUBSAPPL  /STA PGM  &EHKVAR2'),
```

The recovery action definition will occur for all transactions or programs that require recovery.

| **Note:** The IMS/ESA V6 DFSNDMX0 Non-Discardable Message Exit Routine
| will not alter the behaviour of this recovery action.

Using State/Action Tables

State/Action tables provide a formal method and framework to assist in analyzing and defining automation. The automation extensions implemented by the state/action tables are used as input to NetView during execution. For a general discussion of how state/action tables work, see “State/Action Tables” on page 7.

Following is a list of supporting components:

- The common state handler routine that reads the state/action tables and is responsible for the control of the process. This routine is the only routine that updates the states that indicate which action to invoke.
- Default state/action tables for several areas, such as OLDS problem recovery, MSC link recovery, and transaction/program recovery.
- Common routines to be used by the action routines.
- Action routines for several areas.
- Entries in the control file

State/Action Table Control File Entries

In the control file, the PRODUCT and AREA entries define state/action table information.

PRODUCT *product***,SUBSYS=(***subsys***,AREA=***area***)**
Sets of state/action tables are grouped by product. In this case, the product is IMS. SUBSYS= is the subsystem name, and AREA points to the AREA control file entry defined for this subsystem. If different actions are required for different subsystems, then you can create another AREA entry and reference it here with the AREA keyword.

AREA *area***,***function***=***sat-table*
This entry is used to define the specific state/action table that is accessed for each function, where *area* associates this set of definitions with the specific subsystems (the same AREA name is specified in the PRODUCT entry), and *function* is a type of state/action table, such as OLDS, and *sat-table* is the name of the table for that particular state/action table. This is the actual member name.

Example Control File Entries for State/Action Table Definitions

```
PRODUCT  IMS,SUBSYS=(IMS10AA,AREA=IMS),
             SUBSYS=(IMS10AB,AREA=IMS),
             SUBSYS=(IMS10Z,AREA=IMSTEST)
AREA     IMS,MSC=EVISS002,
             OLDS=EVISS003,
             TRAN=EVISS005
AREA     IMSTEST,MSC=EVISS102,
             OLDS=EVISS103,
             TRAN=EVISS105
```

In this example, the AREA= keyword in the PRODUCT control file entry defines which AREA entry to use for each IMS subsystem.

The AREA control file entry provides the names of the state/action tables for each area. The AREAs are:

MSC The state/action table used for MSC link error recovery
OLDS The state/action table used for OLDS error recovery
TRAN The state/action table used for transaction error recovery

In our example, the IMS10Z AREA= keyword in the PRODUCT control file entry specifies IMSTEST. The AREA control file entry for IMSTEST has a different set of tables defined to it.

State/Action Table Keywords

These keywords are used within the state/action tables:

AREA= Contains the AREA within a subsystem that the state/action table reflects (MSC, OLDS, TRAN, ...).

STATE= Contains the initial state value (optional), normally zero. Any other state value may be specified during testing to check the action control of your state/action table.

EVENT= Defines the message for which the actions in the event row apply.

| Should you wish to continue your actions for a single event, over
| more than 1 line, then enter a comma as the last character of the
| line to be continued.

PREPROC= Names a preprocessor that is run before the routines defined in this table are invoked.

Maintaining Security

IMS Automation provides the ability to authorize, by class, those operators that can perform specific functions against specific IMS subsystems. The operator classes are coded in DSICMD, member EVISECUR, by KEYCLASS and VALCLASS. KEYCLASS can be an IMS subsystem name (as defined to IMS Automation), a domain, a group name (also as defined to IMS Automation). VALCLASS identifies functions allowed for the subsystem, group, or domain specified with the previous KEYCLASS entry. See Table 5.

Table 5. Security Entries

Entry		Explanation
IMS10AA	KEYCLASS 1	Operators with class 1 can work with IMS10AA.
INQUIRY	VALCLASS 1	Operators with class 1 can invoke an inquiry on IMS10AA.
STARTUP	VALCLASS 2	Operators with class 2 can invoke a startup on IMS10AA.
SHUTDOWN	VALCLASS 2	Operators with class 2 can invoke a shutdown on IMS10AA.
SERVICEP	VALCLASS 2	Operators with class 2 can work with service periods on IMS10AA.
TRIGGERS	VALCLASS 2	Operators with class 2 can work with triggers on IMS10AA.
BRO	VALCLASS 2	Operators with class 2 can broadcast to IMS10AA.
SUPPORT	VALCLASS 2	Operators with class 2 can work with support function on IMS10AA.
EVIPPICT	VALCLASS 2	Operators with class 2 can start and stop the PPI on IMS10AA.
=OTHER	VALCLASS 3	Operators with class 3 can perform any other function on IMS10AA not already defined in this group of entries (default).
=OTHER	KEYCLASS 9	Operators with class 9 can work with any other IMS.
=OTHER	VALCLASS 9	Operators with class 9 can perform any function on any other IMS.

Notes:

1. Entries are grouped by subsystem, with the subsystem listed as the first item in the group (KEYCLASS). All function items (VALCLASS) following a subsystem item are associated with that subsystem until another subsystem item (KEYCLASS) is met.
2. =OTHER is used as the default entry and can be used for either KEYCLASS or VALCLASS.
3. You can use IMS Automation security checking for your own functions.

See “Adding Local Applications to the IMS Automation Operator Interface” on page 182 for information on adding your local applications to the IMS Automation interface.

Part 2. Programming the IMS Automation Feature

Chapter 3. Programming Information 41
Differences in AOC/MVS Release 1 and SA OS/390 1.3 Syntax 41
Upgrading from IMS Automation 1.3 to IMS Automation 1.4 Using ARM Support 41
Special Considerations when Coding Control File Entries 41
Restrictions 42
Using Multiple Control File Members for Configuration Data 42
Changing Control File Entries 42
Using Partial Reload for Control File Entries 43
Using a Master Control File 43
Notational Conventions and Syntax Rules 44
ABCODEPROG—Respond to BMP region abends 45
ABCODES/ACTCODES/ALTCODES—Restart control region after abend . . . 47
ABCODETRAN—Respond to transaction abend codes 49
ACORESTART—SA OS/390 51
AREA—Define a set of state/action tables 52
AUTOOPS—Define global automation tasks to SA OS/390 53
AUTOOPS—Define automation tasks for IMS subsystems 54
BRO—Broadcast a message prior to shutdown 55
CHE—Issue a checkpoint command 56
| CQS0031A—Confirm CQS restart for structure 57
| CQS0032A—Respond to CQS structure restart 58
| CQS0033A—Respond to client takeover restart 59
| CQSET—Issue Structure Checkpoint at CQS Termination 60
DFS2142—Respond to stopped logical link path message 61
DFS2161I—Link stopped by other system 63
DFS2169I—Respond to MSC link disconnection message 64
DFS3258A—No online data sets available 65
DFS3869A/DFS3869I—Alternate XRF IMS subsystem detects potential failure 66
DFS554A—Respond to program abend 67
DFS810A—Define restart commands 69
DFS989I—Define Restart Commands (DBCTL Only) 71
DFS994I—Respond to checkpoint written to the IMS log 73
ENVIRON—Tailor operations 76
ENVIRON SETUP—Identify startup routine 83
EXTCOND—External conditions 84
FORCE—Takeover by alternate XRF IMS subsystem 86
HOLDQ—Issue commands at shutdown 87
IMSCNTL—IMS/DC control region 88
IMS CRITMSGS—Display critical messages in the SDF 90
IMSGROUP—Group of IMS subsystems 91
IMSRSENM—IMS resource name 92
INITSTART—Control subsystem start 93
IPS—XRF IMS takeover commands 95
NTFYOP—Define notification operator 96
OLDS—Define recovery criteria for OLDS 97
POSTCHKP—Issue commands after IMS shutdown checkpoint 100
PRECHKP—Issue commands prior to IMS shutdown checkpoint 101
PRODUCT—Subsystem state/action table sets 102
RECONS—Set monitoring interval for RECONS 103
RECOVERY—Turn automation on/off 104

RELEASEQ—Issue commands after shutdown completes 106
RESTART—Control restart of IMS subsystem 107
RESTARTABORT—Emergency restart commands 108
SDF support 110
SERVICE—Service period definitions 112
SHUTINIT—Initial shutdown commands 114
SHUTNORM/SHUTIMMED/SHUTFORCE—Control region 115
SHUTNORM/SHUTIMMED/SHUTFORCE—Message, FP, and BMP regions 117
SHUTTYPES—Issue commands at operator shutdown 120
SNAPQ—Issue SNAPQ checkpoint command 123
STARTn—Define start commands 124
STARTUP—Issue commands at IMS subsystem startup 126
STOPBMPREGION—Stop batch message regions 128
STOPFPREGION—Stop fast path regions 129
STOPREGION—Stop IMS dependent message region 130
STSFILE—Status file 131
SUBSYSTEM—Define IMS region to SA OS/390 133
TCO—Issue commands for time-driven procedures 136
TCOMEMBERS—Define TCO Members 137
THRESHOLDS—IMS error thresholds 139
TPABEND—Dependent region abend 141
TRIGGER—Startup and shutdown triggers 142
UNLKAVM—Reply to message AVM005A 144
UNLOCK—Input/output prevention completed 145
| UP—Initiate FDR Startup 146
USERSTART—User-defined restart commands 147
VTAMTERMS—Release terminal sessions 148
%INCLUDE—Include additional members in control file 149
$PI—Stop BMP tasks on active subsystem 150
$SI—Issue start initiator commands after XRF takeover 151
$TI—Assign job classes after XRF takeover 152

Chapter 4. Common Routines 153
EVIEX002—Retrieve XRF partner data 154
EVIEX003—Update status file data 156
EVIEX004—Retrieve XRF partner data 163
IMSBMSG—Build message processor 165
IMSCMD—Issue IMS commands 167
IMSFWM—Add messages to SDF 170
IMSPOST—Post an external event 172
IMSQRY—Name lookup 174
IMSRCMD—Request an IMS function 178
IMSSEC—Invoke security checking 179
EVISROUT—IMS message router 181
EVISROUT—IMS message router (contd.) 182
  Adding Local Applications to the IMS Automation Operator Interface . . . 182

Chapter 3. Programming Information

This section describes the format of IMS Automation control file entries used by the system programmer. The control file identifies the attributes of the system environment, the resources to be automated, and the types of automation activities that will occur. Entries are listed in alphabetical order. Each entry description includes the format and description of operands and, where applicable, usage notes and examples.

Note: All control file entries are **required** unless otherwise specified.

Differences in AOC/MVS Release 1 and SA OS/390 1.3 Syntax

The examples shown in this book follow the syntax used by AOC/MVS Release 4. If you are migrating from AOC/MVS Release 1 to SA OS/390 1.3, change the underscores (used as separators) to periods. For example, this syntax in SA OS/390 1.3:

```
THRESHOLDS IMS10AA.TRAN.SAMPLE1,CRIT=...
```

uses underscores as separators if you use Release 1:

```
THRESHOLDS IMS10AA_TRAN_SAMPLE1,CRIT=...
```

Upgrading from IMS Automation 1.3 to IMS Automation 1.4 Using ARM Support

The IMS Automation customization that is currently used with IMS Automation 1.3 can be used unchanged with IMS Automation 1.4 with ARM support.

However, when you decide to exploit the ARM capability available in MVS/ESA SP Version 5 Release 2 and SA OS/390 1.3, additions are needed. For example,

- The subsystem must be defined to SA OS/390 for each system that is a candidate for an ARM restart. Primary and secondary associations must be specified as well as the ARM element name. More information on customizing SA OS/390 for the ARM environment is found in *SA OS/390 Customization*.
- Likewise, IMS unique customization, such as IMSCNTL, SERVICE, TRIGGER, is needed wherever ARM can restart the subsystem. Including the same member that contains only the SA OS/390 IMS unique information in each ACF file can simplify this step.
- Additional XRF-only customization for secondary system images is needed.

Special Considerations when Coding Control File Entries

This section provides general help information for coding IMS Automation control file entries.

Restrictions

SA OS/390-IMS uses the NetView KEYCLASS statement for security checking of IMS subsystem names. NetView limits the length of a keyword in the KEYCLASS statement to 8 characters. This also limits the length of an IMS subsystem name to 8 characters, contrary to SA OS/390 that supports names with 11 characters.

Using Multiple Control File Members for Configuration Data

If you divide the configuration data into multiple members rather than using a single control file member, use the %INCLUDE statement described in "%INCLUDE—Include additional members in control file" on page 149.

%INCLUDE is also described in the SA OS/390 base documentation.

Note: During initialization, these members are loaded to create a single virtual control file.

Changing Control File Entries

Two types of control file entries define IMS automation:

SA OS/390 control file entries
These entries already exist in SA OS/390. Specific keywords are required and some entries have been extended to support additional IMS automation capabilities.

Those entries unique to IMS Automation
IMS Automation provides additional entries required for IMS automation.

To change the control file entries, you can use either an editor or the SA OS/390 customization dialogs.

If you build with an editor
With SA OS/390 you can build control entries with an editor if you know the control file entries syntax. This book documents the IMS Automation-unique control file entries used by IMS Automation and shows their syntax. Complete documentation for all SA OS/390 control file entries is not included in this manual.

If you build with the customization dialog
Most of the SA OS/390 control file members used for IMS Automation are supported with the SA OS/390 customization dialogs. Consider the following when deciding whether to use the customization dialogs for IMS Automation control file entries:

- %INCLUDE members must be built with an editor.
- Control file members built with the dialog must be always maintained with the dialog. They cannot be edited.

  Notes:

  1. If a member initially built with the dialog is edited, the edited material is lost the next time the dialog is used on that member.
  2. With SA OS/390, the SYSTEM member is built with the dialog as well as the subsystems that are supported by a dialog. **Once you edit a member, you must continue to edit it to maintain it.** If you switch

back to using the dialog, you will lose the information you changed during the edit session.

- Some IMS Automation control file entries are not supported through the customization dialogs.

If you were to look at the control file configuration data, the entries would look similar to those in Table 6.

Table 6. Control File Configuration Data

Configuration data	What it does
`SUBSYSTEM IMS1,JOB=IMSONE,` `PARENT=VTAM`	Defines the symbolic IMS name, the job that starts this IMS, and the parent subsystem.
`IMSCNTL IMS1,APPLID=VTAMAPPL,MAJNODE=VTAM1`	Defines the VTAM name for this IMS and the VTAM node.
`SERVICE IMS1,DAY=(WEEKDAY,0800-1700)`	Defines the service periods in which this IMS will start up and shut down.
Note: All of the examples are shown as they would appear in the control file.	

Using Partial Reload for Control File Entries

If you define a new subsystem in multiple members, then the member containing the SUBSYSTEM configuration data must be the first member loaded by the partial reload function.

Using a Master Control File

It is recommended that you use a master control file that contains nothing but %INCLUDE statements, such as the example shown in Figure 14.

```
***********************************************************************
*    DESCRIPTION:  THE MASTER CONFIGURATION CONTROL FILE              *
***********************************************************************
%INCLUDE SYSTEM
%INCLUDE SETUP
%INCLUDE VTAMTSO
%INCLUDE RMF
%INCLUDE IMSALL
%INCLUDE CICSALL
%INCLUDE STATUS
```

Figure 14. Master Control File Example

In this example master file, the %INCLUDE members, except STATUS, contain configuration data specific to subsystems. STATUS contains the control file entries for SDF. SDF entries must be coded with an editor.

In this example, SYSTEM would contain definitions for notification operators, environment attributes, automation operators, notification forwarding, gateway definitions, and all other configuration information that does not pertain to a specific subsystem.

The example shown in Figure 14 on page 43 is only a sample of how you can set up a master control file. You can name the members anything you like. You can also have separate members for each subsystem, for gateway definitions, for notification operators, and so on. See "Using Multiple Control File Members for Configuration Data" on page 42 for additional information.

If you are using secondary associations, it is recommended that a common definition of SA OS/390 IMS specific customization be used. For example, in Figure 14 on page 43 `IMSALL` should be used in each system in the sysplex.

Notational Conventions and Syntax Rules

The syntax and notational conventions used for the entries are as follows:

- Items shown in braces { } represent alternatives. You must choose one. For example, `{A|B|C}` indicates that you must specify one item only, either A, B, or C.
- Items shown in brackets [ ] are optional. You may choose one. For example, `[A|B|C]` indicates that you may enter A, B, or C, or you may omit the operand.
- An ellipsis (...) indicates that a variable number of items may be included in the list.
- An underscored item shows the default that the system will choose if you do not specify an item. For example, [A| **B**|C] indicates that if no operand is specified, B is assumed.
- Lowercase italicized items are variable; substitute your own value for them.
- Uppercase items must be entered exactly as shown.
- Parentheses must be entered as shown in the syntax diagrams.
- Where operands can be abbreviated, the abbreviations are shown in capital letters. For example, SUBsystem can be entered as `SUB` or `SUBSYTEM`.
- Where brackets ([ ]) are nested, you must include commas to denote the absence of the required positional operands. For example, enter `A,,C` to specify C only for the following command syntax:

  `[A[,B[,C]]]`
- Note that the comma in the example is placed before the actual parameter. This is done to illustrate different parameters. When the entries are actually coded, the comma must be placed after the parameter value.

ABCODEPROG—Respond to BMP region abends

Use this entry to define actions to be taken for program abends of program-driven batch message processing (BMP) regions. Only abends for program-driven BMPs use this entry. Transaction driven BMPs use ABCODETRAN to determine recovery actions.

```
ABCODEPROG subsys.PROG[.progid],
           CODE=(*,acode,progid,INCLUDE|EXCLUDE),
           CODE=(*,acode,progid,INCLUDE|EXCLUDE)
```

Keyword and Parameter Definitions

subsys
The symbolic name by which this IMS subsystem is known to SA OS/390, as defined with the subsystem entry.

PROG[.*progid***]**
The PROG keyword is required. If the PROG keyword is used alone, the entry applies to all the programs. If the program ID is coded after the PROG keyword (PROG.*progid*), the entry applies only to that program.

CODE=
Use this keyword to define which abends should be included or excluded from recovery.

* An asterisk in first positional parameter is required for compatibility with ABCODETRAN entry. Code an asterisk as shown.

acode
The abend code. An asterisk (*) can be used for generic specifications. System abend codes should be prefixed with an S, such as S0C1.

progid
The program name.

INCLUDE|EXCLUDE
Indicates whether or not to initiate a recovery sequence for this program and abend code combination. Use INCLUDE to initiate a recovery and EXCLUDE if you do not want a recovery initiated. If a program and abend combination is specified but INCLUDE|EXCLUDE is omitted, INCLUDE is assumed.

Comments and Usage Notes

If the ABCODEPROG entry is omitted, no recovery takes place and a warning message is issued.

The program name can be specified as *subsys*.PROG.*progid* or as the third parameter in a CODE= statement. Use *subsys*.PROG.*progid* when you want all of the specifications to apply to one specific program. Use CODE=(,,*progid*...) when you want to have a generic entry for the IMS being automated.

Examples of Usage

Example 1

```
ABCODEPROG IMS10AA.PROG,CODE=(*,U0456,*,INCLUDE),
                        CODE=(*,S0C1,SAMPLE1,EXCLUDE),
                        CODE=(*,*,*,INCLUDE)
```

This example uses the PROG keyword alone, so this entry applies to all the programs that do not have an individual entry coded for them.

Example 2

```
ABCODEPROG IMS10AA.PROG.SAMPLE1,
                        CODE=(*,S0C1,*,EXCLUDE)
                        CODE=(*,*,*,INCLUDE)
```

This entry uses the PROG keyword followed by a program ID, SAMPLE1. This entry applies to the SAMPLE1 program only.

ABCODES/ACTCODES/ALTCODES—Restart control region after abend

IMS Automation processing utilizes the SA OS/390 CODE specification for automated error code matching, to specify the abend codes for which SA OS/390 will automatically restart an IMS control region after the abend process is complete.

```
subsys   ABCODES,CODE=({SYScode|IMScode},,,ABENDING|STOPPING),
                 .
                 .
                 .
                 CODE=({SYScode|IMScode},,,ABENDING|STOPPING)
subsys   ACTCODES,CODE=({SYScode|IMScode},,,ABENDING|STOPPING),
                  .
                  .
                  .
                  CODE=({SYScode|IMScode},,,ABENDING|STOPPING)
subsys   ALTCODES,CODE=({SYScode|IMScode},,,ABENDING|STOPPING),
                  .
                  .
                  .
                  CODE=({SYScode|IMScode},,,ABENDING|STOPPING)
```

Keyword and Parameter Definitions

subsys
: This is the SA OS/390 subsystem name of the IMS control region.

ABCODES
: An ABCODES entry lists codes for a non-XRF IMS control region.

ACTCODES
: An ACTCODES entry lists codes for an *active* XRF IMS control region.

ALTCODES
: An ALTCODES entry lists codes for an *alternate* XRF IMS control region.

CODE=
: Use this keyword to define the data, as shown in the following list:

{SYS*code*|IMS*code*}
: Specify the matching abend code here. All abend codes must start with either SYS for MVS system abend codes or IMS for IMS user abend codes. For system abends, *code* is a three-character alphanumeric. For IMS user abends, *code* is a four-digit number.

ABENDING|<u>STOPPING</u>
: ABENDING defines a recoverable IMS abend. With this abend, IMS can be restarted. STOPPING defines a non-recoverable IMS abend.

Comments and Usage Notes

1. If an abend code is omitted, the abend is considered non-recoverable.
2. When an IMS control region abends with an abend code listed in the appropriate table, SA OS/390 will automatically restart the IMS control region.
3. One of the three tables (shown in the following example) is searched depending on the type of IMS control region that is abending. For an XRF IMS subsystem, it is recommended that both ACTCODES and ALTCODES entries be specified.

Examples of Usage

Examples

```
IMSIMSZ    ABCODES,CODE=(SYS0C4,,,ABENDING),
                   CODE=(IMS0113,,,ABENDING),
                   CODE=(IMS0020,,,ABENDING),
                   CODE=(IMS0707,,,ABENDING)

IMSIMSZ    ACTCODES,CODE=(SYS0C4,,,ABENDING),
                    CODE=(IMS0020,,,ABENDING),
                    CODE=(IMS4095,,,ABENDING),
                    CODE=(IMS0590,,,ABENDING)

IMSIMSZ    ALTCODES,CODE=(SYS0C4,,,ABENDING),
                    CODE=(IMS0020,,,ABENDING),
                    CODE=(IMS4095,,,ABENDING)

IMSIMSZ    ALTCODES,CODE=(SYS0C4,,,STOPPING),
                    CODE=(IMS0020,,,STOPPING),
                    CODE=(IMS4095,,,STOPPING)
```

In the last example, you can omit the word STOPPING, because STOPPING is the default value.

ABCODETRAN—Respond to transaction abend codes

Use this entry to define actions to be taken for transaction abend codes.

```
ABCODETRAN subsys.TRAN[.transid],
           CODE=(transid,acode,*,{INCLUDE|EXCLUDE}),
           [CODE=(transid,acode,*,{INCLUDE|EXCLUDE})],
           .
           .
           .
           [CODE=(transid,acode,progid,{INCLUDE|EXCLUDE})        ]
```

Keyword and Parameter Definitions

subsys
The symbolic name by which this IMS subsystem is known to the SA OS/390, as defined with the SUBSYSTEM control file entry.

TRAN[.*transid*]
This entry stands for the IMS transaction recovery which includes both transaction-driven message processing regions (MP) and transaction-driven batch message processing (BMP) regions. The TRAN keyword is required. If the TRAN keyword is used alone, the entry applies to all transactions. If the transaction ID is coded after the TRAN keyword (TRAN.*transid*), the entry applies to that transaction only.

CODE=
Use this keyword to define the data, as shown in the following descriptions:

transid
The transaction ID. An asterisk (*) can be used for generic specifications.

acode
The abend code. An asterisk (*) can be used for generic specifications.

progid
The program name. An asterisk (*) can be used for generic specifications.

INCLUDE|EXCLUDE
Indicates whether or not to initiate a recovery for this transaction and abend code. Use INCLUDE to initiate a recovery and EXCLUDE if you do not want a recovery initiated.

Comments and Usage Notes

1. The transaction name can be specified as TRAN.*transid* or or as the first parameter in the CODE= statement. Use TRAN.*transid* when you want all of the specifications to apply to one specific transaction. Use CODE=(*transid* when you want to code one definition for an IMS subsystem.

Examples of Usage

Example 1

```
ABCODETRAN IMS10AA.TRAN,CODE=(PART,U3303,DFSSAM04,INCLUDE),
                     CODE=(*,U3303,*,EXCLUDE),
                     CODE=(*,*,*,INCLUDE)
```

This example uses the TRAN keyword alone, so this entry applies to all the transactions that do not have an individual entry coded for them.

Note: The actions coded on the ABCODETRAN entry are processed in the order they are coded.

Example 2

```
ABCODETRAN  IMS10AA.TRAN.PART,
            CODE=(*,U3303,DFSSAMOY,INCLUDE),
            CODE=(*,*,*,EXCLUDE)
```

This entry uses the TRAN keyword followed by a transaction ID, PART. This entry applies to the PART transaction only.

ACORESTART—SA OS/390

An ACORESTART entry defines commands to be executed when SA OS/390 is started or stopped and restarted with automated subsystems that are already active. An ACORESTART entry must be coded for each IMS control region.

Note: ACORESTART is an SA OS/390 control file entry. The associated SA OS/390 customization dialog menu selection is AOC RESTART.

```
subsys    ACORESTART,CMD=(,,'EVIEE00A subsystem')
```

Keyword and Parameter Definitions

EVIEE00A
EVIEE00A identifies the CLIST required by IMS Automation for initialization processing after each SA OS/390 startup. Code exactly as shown.

subsystem
The subsystem name is **required**.

Comments and Usage Notes

⇒ **Required**. Code exactly as shown.

This specification will allow IMS Automation to do initialization processing after each SA OS/390 startup.

For each active IMS control region defined to IMS, a /DISPLAY ACTIVE command will be issued. If the status file reflects a status other than the one retrieved from the IMS display command, the status file will be updated appropriately.

If any dependent regions are found inactive, IMS Automation will send a notification to the operator. Furthermore, the monitoring for RECONS data set exceptions and the monitoring for online data sets (OLDS) will be started.

Examples of Usage

Example

```
IMSIMSZ    ACORESTART,CMD=(,,'EVIEE00A IMSIMSZ')
```

AREA—Define a set of state/action tables

Use this entry to define a set of state/action tables.

```
AREA type,MSC=table_name,
          OLDS=table_name,
          TRAN=table_name
```

Keyword and Parameter Definitions

type
Associates this set of tables with a subsystem, as defined in the control file entry described in “PRODUCT—Subsystem state/action table sets” on page 102.

MSC=
This entry provides the name of the table that defines automatic recovery from MSC link errors.

OLDS=
This entry provides the name of the table that defines automatic recovery procedures for OLDS errors.

TRAN=
This entry provides the name of the table that defines automatic recovery procedures initiated when a MPP or BMP transaction/program has abended.

Comments and Usage Notes

A PRODUCT entry must be defined. See “PRODUCT—Subsystem state/action table sets” on page 102.

Examples of Usage

Example 1

```
AREA     IMS,MSC=EVISS002,
             OLDS=EVISS003,
             TRAN=EVISS005
```

These are the default tables.

Example 2

```
AREA     IMS1,MSC=EVISS102,
             OLDS=EVISS103,
             TRAN=EVISS105
```

In this case, different tables are defined for subsystem IMS1.

AUTOOPS—Define global automation tasks to SA OS/390

The purpose of the AUTOOPS entry is to define an IMS Automation automation task to SA OS/390. The difference between the AUTOOPS subsystem entry (refer to “AUTOOPS—Define automation tasks for IMS subsystems” on page 54) and the SA OS/390 global entry is that the subsystem entry defines an automation task for individual IMS subsystems, while the SA OS/390 global entry defines automation tasks assigned to all IMS subsystems. The three AUTOOPS (SA OS/390) entries shown below must be coded to define three required SA OS/390 automation tasks.

```
AUTOOPS   IMSMSTR,ID=(auto),
                MSG=(DFS*,AVM005*,AVM006*,IOS071*,DSP*,EVI*,DXR*)
          AUTOOPS    IMSWATCH,ID=(auto)
          AUTOOPS    IMSPPI,ID=(auto)
```

Keyword and Parameter Definitions

ID=*auto*
The actual NetView automation task name. While *auto* can be locally defined to be any unused, validly-defined automation task, we recommend that you use the following wherever possible:

- AUTIMS for the IMSMSTR AUTOOPS
- AUTSURV for the IMSWATCH AUTOOPS
- AUTIPPI for the IMSPPI AUTOOPS

The recommended names are used in the examples.

MSG=
Defines sets of messages to be trapped and routed to IMS subsystems identified in AUTOOPS subsystem entries (refer to “AUTOOPS—Define automation tasks for IMS subsystems” on page 54).

Comments and Usage Notes

⇒ **Required**.

These three entries are coded for any IMS Automation configuration and will appear only once in your control file member regardless of the number of IMS subsystems defined. They provide the base automation tasks for IMS message processing.

Examples of Usage

Examples

```
AUTOOPS    IMSMSTR,ID=(AUTIMS),
           MSG=(DFS*,AVM005*,AVM006*,IOS071*,DSP*,EVI*,DXR*)

AUTOOPS    IMSWATCH,ID=(AUTSURV)

AUTOOPS    IMSPPI,ID=(AUTIPPI)
```

AUTOOPS—Define automation tasks for IMS subsystems

These additional AUTOOPS entries are required to support individual IMS subsystems as they are added to SA OS/390. Each IMS subsystem must have an associated AUTOOPS automation task defined to SA OS/390.

```
AUTOOPS    IMSOPnn,ID=(operator_id)
```

Keyword and Parameter Definitions

IMSOP*nn*
: IMSOPnn can be any characters as long as the string matches the AUTOOPS keyword value on the IMSCNTL entry. See “IMSCNTL—IMS/DC control region” on page 88 for additional information.

operator_id
: The automation task names used can be any locally defined automation task name.

Comments and Usage Notes

Each IMS control region defined to and controlled by SA OS/390 must have an additional AUTOOPS entry defined as described.

Examples of Usage

An example follows of the AUTOOPS entries required to define three IMS control regions.

Example

```
AUTOOPS    IMSOP01,ID=(AUTIMSZ)
AUTOOPS    IMSOP02,ID=(AUTIMSA)
AUTOOPS    IMSOP03,ID=(AUTIMSB)
```

BRO—Broadcast a message prior to shutdown

This entry defines the command and message text that may be issued prior to normal shutdown to indicate to active users that shutdown of the specified IMS subsystem is imminent.

```
subsys  BRO,REPLY=(,RETRYn,'/BRO ACTIVE')
          [,REPLY=(,RETRYn,'message text.')]
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

/BRO ACTIVE
This command may be issued prior to normal shutdown to indicate to active users that the IMS subsystem (*subsys*) shutdown is imminent.

message text
The message text may be issued by the /BRO ACTIVE command as described in the parameters above. The variable &EHKVAR1 can be used in the message text; its value will be the number of minutes that IMS Automation will wait until the IMS is shut down, as specified in the BTIMER parameter of the ENVIRON entry (refer to “ENVIRON—Tailor operations” on page 76).

Comments and Usage Notes

⇒ **Not required**. Not applicable for DB control regions.

You must end the *message text* input for this entry with a period.

This IMS command and associated message text can be issued just prior to the normal IMS control region shutdown process.

One variable is available for use in the BRO entry. &EHKVAR1 is the Broadcast value specified in the control file subsystem ENVIRON entry.

Examples of Usage

Example

```
IMSIMSZ  BRO,REPLY=(,RETRY5,'/BRO ACTIVE'),
             REPLY=(,RETRY5,'IMS SHUTTING DOWN IN &EHKVAR1 MINUTES.')
```

CHE—Issue a checkpoint command

This entry specifies the IMS command to be issued when a normal checkpoint is required.

```
IMS Control Region Syntax:
subsys  CHE,REPLY=(,RETRYn,'/CHE')

DB Control Region Syntax:
subsys  CHE,CMD=(,,'MVS &EHKVAR7CHE')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

RETRY*n*
The word RETRY must be followed immediately by the number of retry attempts (*n*).

/CHE
Code the /CHE command exactly as specified.

Comments and Usage Notes

This IMS command is issued when a normal checkpoint is required.

&EHKVAR7 is a variable containing the console command character.

Examples of Usage

Example 1

```
IMSIMSZ CHE,REPLY=(,RETRY9,'/CHE')
```

Example 2

```
IMS01C CHE,CMD=(,,'&EHKVAR7CHE')
```

|
CQS0031A—Confirm CQS restart for structure

This entry defines the reply required in response to the WTOR message CQS0031A during CQS startup.

```
subsys  CQS0031A,REPLY=(,,'CONFIRM')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the CQS region.

Comments and Usage Notes

Not applicable to non-shared queue systems.

This reply is issued in response to CQS0031A WTOR informing subsys that CQS could not read the system checkpoint log token from the checkpoint data set, but found a system checkpoint log token in the control entry in the shared queues structure. Replying CONFIRM allows CQS to use the log token in the CQS0031A message (i.e. the system checkpoint log token in the control entry) for restart processing.

Examples of Usage

Example

```
IMS10C CQS0031A,REPLY=(,,'CONFIRM')
```

CQS0032A—Respond to CQS structure restart

This entry defines the reply required in response to the WTOR message CQS0032A during CQS startup.

```
subsys  CQS0032A,REPLY=(,,'COLD')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the CQS region.

Comments and Usage Notes

Not applicable to non-shared queue systems.

This reply is issued in response to CQS0032A WTOR informing subsys that CQS could not read the system checkpoint log token from either the checkpoint data set or from the control entry in the shared queues structure. CQS does not know where to start reading from the log without the log token. Replying COLD allows CQS to begin coldstart processing.

Examples of Usage

Example

```
IMS10C CQS0032A,REPLY=(,,'COLD')
```

CQS0033A—Respond to client takeover restart

This entry defines the reply required in response to the WTOR message CQS0033A during CQS startup.

```
subsys  CQS0033A,REPLY=(,,'COLD')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the CQS region.

Comments and Usage Notes

Not applicable to non-shared queue systems.

This reply is issued in response to CQS0033A WTOR informing subsys that CQS could not read the system checkpoint log token for the CQS that was connected to the failed client from the control entry in the shared queues structure or an error prevented CQS from accessing all required log records. Replying COLD allows CQS to begin client takeover coldstart processing.

Examples of Usage

Example

```
IMS10C CQS0033A,REPLY=(,,'COLD')
```

CQSET—Issue Structure Checkpoint at CQS Termination

This entry defines the /CQSET command that will be issued during IMS startup to cause a Structure Checkpoint at CQS shutdown.

```
subsys CQSET,CMD=(,,'IMSCMD &SUBSAPPL /CQSET SHUTDOWN SHAREDQ ON STRUCTURE ALL')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the CQS region.

Comments and Usage Notes

Not applicable to non-shared queue systems.

The command is issued during IMS startup. The process is triggered by message EVI130I indicating that the PPI is active. When the /CQSET command is issued, IMS passes the request to CQS when IMS terminates normally with a /CHECKPOINT FREEZE|DUMPQ|PURGE command.

IMS Feature code will ensure that the &SUBSAPPL task global is set to the correct IMS.

IMS Feature code correlates CQS regions to IMS Control regions during IMS Control region startup.

Examples of Usage

Example

```
IMS10C  CQSET,CMD=(,,'IMSCMD &SUBSAPPL /CQSET SHUTDOWN SHAREDQ ON STRUCTURE ALL')
```

DFS2142—Respond to stopped logical link path message

This entry defines the commands required for response to the DFS2142 message following notification that a logical link path was stopped.

```
subsys DFS2142,REPLY=(START,,'/START MSNAME&EHKVAR2')
          [,REPLY=(RESTART,,'/PSTOP LINK   &EHKVAR1')]
          [,REPLY=(RESTART,,'/START MSNAME &EHKVAR2')]
          [,REPLY=(RESTART,,'/RSTART LINK  &EHKVAR1')]
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

START
When message DFS2142 is issued, the START reply will be issued to start the logical link path.

RESTART
When message DFS2142 is issued, the RESTART reply may optionally be issued to stop the link, start the link path and then restart the link.

/START MSNAME &EHKVAR2
Code the /START MSNAME command exactly as specified.

/PSTOP LINK &EHKVAR1
Code the /PSTOP LINK command exactly as specified.

/RSTART LINK &EHKVAR1
Code the /RSTART LINK command exactly as specified.

Comments and Usage Notes

Not applicable for DB control regions.

This reply is issued in response to the DFS2142 message informing *subsys* that a logical link path has stopped. Issuing the /START MSNAME command will activate the link path back. For persistent occurrences of this message, stopping the link, starting the link path and then restarting the link is indicated. If this is done, it must be performed on both IMS systems.

Two variables are used to automate this response.

- &EHKVAR1 is a variable containing the logical link number.
- &EHKVAR2 is a variable containing the logical link path name.

Examples of Usage

Example

```
IMSIMSZ DFS2142,REPLY=(START,,'/START MSNAME &EHKVAR2'),
                REPLY=(RESTART,,'/PSTOP LINK &EHKVAR1'),
                REPLY=(RESTART,,'/START MSNAME &EHKVAR2'),
                REPLY=(RESTART,,'/RSTART LINK &EHKVAR1')
```

DFS2161I—Link stopped by other system

This entry defines the command required (for response to the DFS2161I message) following the stopping of a link by the other system attached to it.

```
subsys DFS2161I,REPLY=(RESTART,,'/RSTART LINK &EHKVAR1')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

RESTART
When message DFS2161I is issued, the RESTART reply will be issued to restart the link with the other system.

/RSTART LINK &EHKVAR1
Code the /RSTART LINK command exactly as shown.

Comments and Usage Notes

Not applicable for DB control regions.

This reply is issued in response to the DFS2161I message informing *subsys* that a link was stopped by the IMS system at its other end. The /RSTART LINK command must be issued on both IMS systems in order to restart the link.

&EHKVAR1 is a variable containing the logical link number.

Examples of Usage

Example

```
IMSIMSZ DFS2161I,REPLY=(RESTART,,'/RSTART LINK &EHKVAR1')
```

DFS2169I—Respond to MSC link disconnection message

This entry defines the command required (for response to the DFS2169I message) following the completion of the disconection of a Multiple Systems Coupling (MSC) link between two IMS systems.

```
subsys DFS2169I,REPLY=(RESTART,,'/RSTART LINK &EHKVAR1')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

RESTART
When message DFS2169I is issued, the RESTART reply will be issued to restart the link with the other system.

/RSTART LINK &EHKVAR1
Code the /RSTART LINK command exactly as shown.

Comments and Usage Notes

Not applicable for DB control regions.

This reply is issued in response to the DFS2169I message informing *subsys* that disconnection of a link is complete. The /RSTART LINK command must be issued on both IMS systems in order to restart the link.

&EHKVAR1 is a variable containing the logical link number.

Examples of Usage

Example

```
IMSIMSZ DFS2169I,REPLY=(RESTART,,'/RSTART LINK &EHKVAR1')
```

DFS3258A—No online data sets available

This entry defines the commands required (for response to the DFS3258A message) following the use of either the last OLDS or all OLDS.

```
subsys DFS3258A,CMD=(SYSTEM,,'command_1'),
                CMD=(LAST,,'command_2')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

SYSTEM
When message DFS3258A is issued and there are no more available OLDS for this IMS subsystem, the SYSTEM command will be issued.

command_1
The command to be executed when there are no more available OLDS. An example of such a command is:

```
IMSCMD &SUBAPPL /LOG WAITING
```

LAST
When message DFS3258A is issued and the IMS subsystem is using the last available OLDS, the LAST command will be issued.

command_2
The command to be executed when the last available OLDS is in use. An example of such a command is:

```
IMSCMD &SUBAPPL /STA OLDS 99
```

Comments and Usage Notes

This command is issued in response to the DFS3258A message.

Examples of Usage

Example

```
IMSIMSZ DFS3258A,CMD=(SYSTEM,,'IMSCMD &SUBSAPPL /LOG WAITING'),
                 CMD=(LAST,,'IMSCMD &SUBSAPPL /STA OLDS 99')
```

DFS3869A/DFS3869I—Alternate XRF IMS subsystem detects potential failure

This entry defines the commands required (for response to DFS3869A or DFS3869I messages) following the *alternate* XRF IMS subsystem's detection of a potential failure of the *active* XRF IMS subsystem.

```
subsys  DFS3869A,REPLY=(,RETRYn,'/SWI SYSTEM FORCE')
subsys  DFS3869I,REPLY=(,RETRYn,'/SWI SYSTEM FORCE')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

/SWI SYSTEM FORCE
This parameter must be specified exactly as shown to enable takeover to proceed.

Comments and Usage Notes

Required only for an XRF IMS subsystem.

These replies are issued in response to messages DFS3869A and DFS3869I, indicating that the *alternate* IMS subsystem has detected a potential failure of the *active* IMS subsystem. REPLY is coded as shown in the format above.

TKO=YES must be specified in "ENVIRON—Tailor operations" on page 76 before the command will be issued.

Examples of Usage

Example

```
IMSIMSZ    DFS3869A,REPLY=(,RETRY5,'/SWI SYSTEM FORCE.')

IMSIMSZ    DFS3869I,REPLY=(,RETRY5,'/SWI SYSTEM FORCE.')
```

DFS554A—Respond to program abend

This entry defines the commands required for response to the DFS554A message, following notification of a program abend.

```
subsys DFS554A,CMD=(TRAN,,'IMSCMD &SUBSAPPL /STA TRAN &EHKVAR1'),
               CMD=(PROG,,'IMSCMD &SUBSAPPL /STA PGM &EHKVAR2')

or

subsys DFS554A,REPLY=(TRAN,,'/STA TRAN &EHKVAR1')
               REPLY=(PROG,,'/STA PGM &EHKVAR2'),
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

TRAN
When message DFS554A is issued and the DFS554A message indicates the transaction is stopped, the TRAN reply or command is issued to restart the transaction.

PROG
When message DFS554A is issued and the DFS554A message indicates the program is stopped, the PROG reply or command is issued to restart the program.

/STA TRAN &EHKVAR1
Code the /STA TRAN command exactly as specified if you want IMS Automation to automatically start the transaction and you are using REPLY.

IMSCMD &SUBSAPPL /STA TRAN &EHKVAR1
Code this command exactly as specified if you want IMS Automation to automatically start the transaction and you are using CMD.

/STA PROG &EHKVAR2
Code the /STA PROG command exactly as specified if you want IMS Automation to automatically start the transaction and you are using REPLY.

IMSCMD &SUBSAPPL /STA PROG &EHKVAR2
Code this command exactly as specified if you are using CMD.

Comments and Usage Notes

For DB control regions, use the **CMD=** syntax only.

This reply is issued in response to the DFS554A message issued when an IMS program abends to restart the transaction and the program.

Two variables are used to automate this response:

- &EHKVAR1 is a variable containing the transaction name.
- &EHKVAR2 is a variable containing the program name.

Four additional variables are supplied for job- or region-unique recovery that may be necessary:

- &EHKVAR3 is a variable containing job identifier.
- &EHKVAR4 is a variable containing region identifier.
- &EHKVAR5 is a variable containing user abend code.
- &EHKVAR6 is a variable containing system abend code.

Examples of Usage

Example 1

```
IMSIMSZ DFS554A,CMD=(TRAN,,'IMSCMD &SUBSAPPL /STA TRAN &EHKVAR1'),
                CMD=(PROG,,'IMSCMD &SUBSAPPL /STA PGM &EHKVAR2')
```

Example 2

```
IMSIMSZ DFS554A,CMD=(TRAN,,'USER REXX &EHKVAR1 &EHKVAR3')
```

In this example, a user-written recovery routine will use the job ID and transaction ID to determine the recovery actions.

DFS810A—Define restart commands

This entry defines the restart commands required (for response to the DFS810A message) following successful completion of IMS initialization.

```
subsys  DFS810A,REPLY=(COLD,RETRYn,'command'),
                REPLY=(BUILDQ,RETRYn,'command'),
                REPLY=(WARMSDBL,RETRYn,'command')
                REPLY=(MANUAL,RETRYn,'&EHKVAR1')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

COLD
When the operator selects a start type of COLD from the Start Functions panel, this parameter is required to define the restart command to be issued in response to message DFS810A, and the number of RETRY attempts before determining that a reply cannot be issued.

BUILDQ
When the operator selects a start type of BUILDQ from the Start Functions panel, this parameter is required to define the restart command to be issued in response to message DFS810A, and the number of RETRY attempts before determining that a reply cannot be issued.

WARMSDBL
When the operator selects a start type of WARMSDBL from the Start Functions panel (causing the MSDB to be loaded during WARM start), this parameter is required to define the restart command to be issued in response to message DFS810A, and the number of RETRY attempts before determining that a reply cannot be issued.

MANUAL
When the operator selects a start type of MANUAL from the Start Functions panel, the Manual Restart pop-up box is invoked. The operator enters an IMS restart command (which is stored in the variable &EHKVAR1). This parameter defines the command to be issued (&EHKVAR1) in response to message DFS810A and defines the number of RETRY attempts before determining that a reply cannot be issued.

command
The IMS command to be issued in response to the DFS810A message. Refer to the example below for sample *command* entries.

&EHKVAR1
This variable contains the IMS restart command entered by the operator on the Manual Restart pop-up box.

Comments and Usage Notes

Not applicable for DB control regions.

These replies are issued in response to the DFS810A message requesting that an /NRESTART or /RESTART be entered.

The REPLY=(MANUAL parameter will normally be coded as indicated in the example.

Examples of Usage

Example

```
IMSIMSZ  DFS810A,REPLY=(COLD,RETRY5,'/NRE CHKPT 0 FORMAT ALL DETACH'),
                 REPLY=(BUILDQ,RETRY5,'/NRE FORMAT RS BUILDQ'),
                 REPLY=(MANUAL,RETRY5,'&EHKVAR1'),
                 REPLY=(WARMSDBL,RETRY9,'/NRE MSDBLOAD')
```

DFS989I—Define Restart Commands (DBCTL Only)

This entry defines the restart commands required (for response to the DFS989I message) following successful completion of IMS initialization.

```
subsys DFS989I,CMD=(AUTO,,MVS command'),
               CMD=(BUILDQ,,'MVS command'),
               CMD=(COLD,,'MVS command'),
               CMD=(MANUAL,,'&EHKVAR1')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

AUTO
When the operator selects a start type of AUTO from the Start Functions panel, this parameter is required to define the restart command to be issued in response to message DFS989I.

BUILDQ
When the operator selects a start type of BUILDQ from the Start Functions panel, this parameter is required to define the restart command to be issued in response to message DFS989I.

COLD
When the operator selects a start type of COLD from the Start Functions panel, this parameter is required to define the restart command to be issued in response to message DFS989I.

MANUAL
When the operator selects a start type of MANUAL from the Start Functions panel, the Manual Restart pop-up box is invoked. The operator enters an IMS restart command (which is stored in the variable &EHKVAR1). This parameter defines the command to be issued (&EHKVAR1) in response to message DFS989I.

command
The IMS command to be issued in response to the DFS989I message. Refer to the example below for sample *command* entries.

&EHKVAR1
This variable contains the IMS restart command entered by the operator on the Manual Restart pop-up box.

Comments and Usage Notes

These commands are issued in response to the DFS989I message requesting that an restart command be entered. For a DBCTL region the entry for message DFS810A is not required since a DBCTL region issues message DFS989I instead of DFS180A.

The **CMD=(MANUAL** parameter will normally be coded as indicated in the example.

The variable &EHKVAR7 contains the IMS command control character.

Examples of Usage

Example

```
IMSIMSZ DFS989I,CMD=(AUTO,,'MVS &EHKVAR7NRE.'),
                CMD=(BUILDQ,,'MVS &EHKVAR7NRE FORMAT RS BUILDQ'),
                CMD=(COLD,,'MVS &EHKVAR7NRE CHKPT 0 FORMAT ALL'),
                CMD=(MANUAL,,'&EHKVAR1')
```

Note: The MANUAL entry can also be defined as:

```
IMSIMSZ DFS989I,CMD=(AUTO,,'MVS &EHKVAR7NRE.'),
                CMD=(BUILDQ,,'MVS &EHKVAR7NRE FORMAT RS BUILDQ'),
                CMD=(COLD,,'MVS &EHKVAR7NRE CHKPT 0 FORMAT ALL'),
                CMD=(MANUAL,,'MVS &EHKVAR7&EHKVAR1')
```

Otherwise, the operator will have to enter the MVS and the IMS command control character.

DFS994I—Respond to checkpoint written to the IMS log

This entry defines the commands required (for response to the DFS994I “*xxxx* START COMPLETED” message) which follows the writing of a checkpoint to the IMS system log.

```
IMS Control Region Syntax:
subsys  DFS994I,REPLY=(ERE,RETRYn,'command'),
                REPLY=(COLD,RETRYn,'command'),
                REPLY=(BUILDQ,RETRYn,'command'),
                REPLY=(WARMSDBL,RETRYn,'/STA DC'),
                REPLY=(WARMSDBL,RETRYn,'/CHE SNAPQ'),
                REPLY=(MANUAL,RETRYn,'command'),
                REPLY=(WARM,RETRYn,'/STA DC'),
                REPLY=(WARM,RETRYn,'/CHE SNAPQ'),
                CMD=(WARM,,'command')

DB Control Region Syntax:
subsys  DFS994I,CMD=(COLD,,'command'),
                CMD=(ERE,,'command'),
                CMD=(MANUAL,,'command'),
                CMD=(WARM,,'command')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

ERE
When the operator selects a start type of AUTO from the Start Functions panel and IMS Automation determines that an emergency restart is necessary, this parameter is required to define the command to be issued in response to message DFS994I, and the number of RETRY attempts before determining that a reply cannot be issued.

COLD
When the operator selects a start type of COLD from the Start Functions panel, this parameter is required to define a command to be issued in response to message DFS994I, and the number of RETRY attempts before determining that a reply cannot be issued.

BUILDQ
When the operator selects a start type of BUILDQ from the Start Functions panel, this parameter is required to define a command to be issued in response to message DFS994I, and the number of RETRY attempts before determining that a reply cannot be issued.

WARMSDBL
When the operator selects a start type of WARMSDBL from the Start Functions panel (causing the MSDB to be loaded during WARM start), this parameter is required to define a command to be issued in response to message DFS994I, and the number of RETRY attempts before determining that a reply cannot be issued.

MANUAL
When the operator selects a start type of MANUAL from the Start Functions panel, a pop-up box displays where the operator enters an IMS restart command. This parameter is required to define the command to be issued in response to message DFS994I and to define the number of RETRY attempts before determining that a reply cannot be issued.

command
The IMS command to be issued in response to the DFS994I message. Refer to the example below for sample *command* entries.

WARM
When the operator selects a start type of AUTO from the Start Functions panel and IMS Automation determines that an warm start is necessary, this parameter is required to define the command to be issued in response to message DFS994I, and the number of RETRY attempts before determining that a reply cannot be issued.

Comments and Usage Notes

These IMS commands and replies are issued after the DFS994I message is received. There may be more than one REPLY to be issued for each type.

Based on the type of restart (warm versus emergency) and the type of startup requested (AUTO, COLD, BUILDQ, and so on), one of the reply types will be selected by IMS Automation and the IMS commands will be issued upon receipt of the DFS994 message. WARMSDBL is used to load the MSDB during a warm start. The WARM and WARMSDBL parameters coded on the DFS994 entry must be the same.

The replies may be specified in any order, but individual parameter types must be coded in the order in which you wish the commands to be issued. For example, for parameter type WARMSDBL, the /STA DC entry must be coded prior to the /CHE SNAPQ entry.

For the keyword REPLY=...'*command*', single quotes are required. For the keyword COMMAND=...'*command*', the single quotes are optional unless the command contains embedded blanks.

Examples of Usage

Example 1 IMS Control Region

```
IMSIMSZ  DFS994I,REPLY=(ERE,RETRY5,'/STA DC.'),
                 REPLY=(ERE,RETRY5,'/CHE SNAPQ.'),
                 REPLY=(ERE,RETRY5,'/STA TRAN ALL.'),
                 REPLY=(COLD,RETRY5,'/STA DC'.),
                 REPLY=(COLD,RETRY5,'/CHE SNAPQ.'),
                 REPLY=(COLD,RETRY5,'/STA NODE ALL.'),
                 REPLY=(BUILDQ,RETRY5,'/STA DC'.),
                 REPLY=(BUILDQ,RETRY5,'/CHE SNAPQ.'),
                 REPLY=(WARMSDBL,RETRY9,/STA DC.'),
                 REPLY=(WARMSDBL,RETRY9,/CHE SNAPQ.'),
                 REPLY=(MANUAL,RETRY5,'/STA DC.'),
                 REPLY=(MANUAL,RETRY5,'/CHE SNAPQ.'),
                 REPLY=(WARM,RETRY5,'/STA DC.'),
                 REPLY=(WARM,RETRY5,'/CHE SNAPQ.'),
                 CMD=(WARM,,USERCMD)
```

Example 2 DBCTL Control Region

```
IMS01C  DFS994I,
          CMD=(COLD,,'MVS &EHKVAR7CHE'),
          CMD=(ERE,,'MVS &EHKVAR7CHE'),
          CMD=(MANUAL,,'MVS &EHKVAR7CHE'),
          CMD=(WARM,,'MVS &EHKVAR7CHE')
```

ENVIRON—Tailor operations

The ENVIRON entry serves as the main facility to tailor the operation of IMS Automation. The ENVIRON entry should be coded for each region which is a part of an IMS system.

Many of the keyword parameters are not required for non-XRF systems. The syntax specification indicates those keyword parameters required for any IMS control region definition. When defining an XRF IMS subsystem, refer to the description of individual parameters to determine the requirements.

```
subsys  ENVIRON
                ,SUBTYPE={CTL|TP|DBRC|DLS|FP|BMP|OTHER}
                [,SHUTGO={YES|NO}]
                [,SUBID=subid]
                [,XRF={NO|YES}]
                [,DEFSTART={AUTO|COLD|BUILDQ|WARMSDBL}]
                [,DEFFORCE={DUMP|NODUMP}]
                [,SYNCH={Y|N|P|S|T}]
                [,DEFXRFST={XRF|ACTIVE|BACKUP}]
                [,APPLVER={YES|NO}]
                [,SDELAY=45|time]
                [,WDELAY=15|time]
                [,BTIMER=3|time]
                [,FTIMER=3|time]
                [,ITIMER=3|time]
                [,RTIMER=3|time]
                [,TTIMER=3|time]
                [,WTIMER=3|time]
                [,ALTSTART={YES|NO}]
                [,ALTRST={YES|NO}]
                [,TKO={YES|NO}]
                [,PRTNRID=XRF partner subid]
                [,PRTNRDOM=XRF or FDR partner domain]
                [,PRTNRSUB=XRF or FDR partner subsys]
                [,WTCHNOTF=60|time]
                [,EXITS=(user_exit)]
                [,CTLNAME=name]
                [,STRTTIMER = 5|time]
                [,DBCTL = YES|NO]
```

Keyword and Parameter Definitions

subsys is the SA OS/390 subsystem name. It is the same name specified on the SUBSYSTEM entry.

SUBTYPE=*IMS subsystem type*

This parameter is the only required entry. For the ENVIRON SUBTYPE entry, specify:

CTL IMS control region
TP IMS message region
DBRC DBRC region
DLS DLISAS region

FP IMS fast path region
BMP IMS batch message processing region
OTHER Other MVS systems that are direct dependents of an IMS control region

SHUTGO={YES|NO}
This parameter is required for the control region only. Specifying YES will allow shutdown of a control region to progress from a normal shutdown, then to an immediate shutdown, then to a forced shutdown (as needed) without any operator action.

SUBID=*subid*
This parameter is required for the control region only. It specifies the subsystem ID (IMSID) specified in the IMS control region parameters or SYSGEN.

XRF={NO|YES}
This parameter is required for the control region only. It specifies whether the control region will be executing as an XRF complex. If the control region when started is currently not XRF-capable and the value specified here is YES, this situation will be recognized by IMS Automation. However, if the control region is started as an XRF IMS subsystem and this value is NO, IMS Automation will not perform correctly to support an XRF complex. It is essential that this parameter be specified to correctly indicate your intentions to IMS Automation.

| **Note:** Specifying XRF=NO in conjunction with the PRTNRDOM= and
| PRTNRSUB= keyword parameters indicates an FDR enabled IMS.

DEFSTART={AUTO|COLD|BUILDQ|WARMSDBL}
This defines the default start type for the Start Functions panel, to be used for a control region startup. This start type will be used when SA OS/390 determines that the subsystem should be running but is not. If SA OS/390 SETSTATE command is issued to perform a start of a subsystem, the default start type will be used. If a default is not specified, the start will fail.

DEFFORCE={DUMP|NODUMP}
This parameter is used during an escalation of shutdown from type IMMED to type FORCE. It retrieves the default shut options for a type FORCE shutdown. This parameter specifies the dump option associated with a type FORCE shutdown.

SYNCH={Y|S|N|P|T}
This parameter is used for XRF systems only and controls the type and degree of region synchronization between the *active* and the *alternate* IMS subsystems. This parameter is effective when the parent IMS control region is the *alternate*. While the parent is the *active* IMS, the parameter is not used.

Value	**Effect**
Y	When the partner region is stopped or started on the *active* IMS subsystem, this region is also stopped or started on the *alternate* IMS subsystem to mirror the *active*.
N	This region will not participate in any type of synchronization.
P	This region will not participate in any type of synchronization; however, it will be started when the *alternate* IMS is started.

S When the partner region is started on the *active* IMS subsystem, this event is recorded. The region will be started on the *alternate* when a takeover occurs after takeover processing is complete.

T This region will be started after the IMS Takeover.

DEFXRFST={XRF|ACTIVE|BACKUP}]
This parameter is used for XRF systems only and it defines the default XRF start type for the Control Region Startup panel, to be used for a control region startup.

APPLVER={YES|NO}]
During IMS startup, if APPLVER is specified as YES, then the IMS control region will not be started unless the APPLID is connectible. In any case, an XRF *alternate* subsystem will not be started regardless of what is specified if the APPLID is not connectible.

SDELAY=45|*time*
This parameter is used for XRF systems only. During shutdown of an XRF IMS complex with an *alternate* that is executing, IMS Automation will attempt the shutdown of the *alternate* prior to the *active*. For an “Immediate” type of shutdown, SDELAY is the time in seconds to wait after initiating *alternate* shutdown before continuing with the *active* shutdown process.

WDELAY=15|*time*
This parameter is used for XRF systems only. During XRF takeover processing, this time is the number of seconds between each invocation of takeover **watch** or monitor processing.

BTIMER=3|*time*
If the BROADCAST or ABORT SHUTDOWN option is selected on one of the shutdown panels, this parameter defines the time in minutes between the execution of the /BRO command and the start of the shutdown (SHUTSYS) process.

FTIMER=3|*time*
This is used to specify the escalation interval in minutes between the transfer from the shutdown type of IMMED to the shutdown type of FORCE. The value specified in this timer overrides the value specified in the TTIMER parameter.

ITIMER=3|*time*
This is used to specify the escalation interval in minutes between the transfer from the shutdown type of NORM to the shutdown type of IMMED. The value specified in this timer overrides the value specified in the TTIMER parameter.

RTIMER=3|*time*
After the first indication of an IMS control region abend (receipt of the DFS629 message), the number of minutes before the final abend recovery routine is invoked.

TTIMER=3|*time*
After the IMS shutdown checkpoint command is issued, the number of minutes before the shutdown verification process is invoked.

WTIMER=3|*time*
This parameter is used for XRF systems only. During an XRF takeover, once the UNLOCK command has been issued by the IMS Automation Surveillance processing this parameter specifies the number of minutes to suspend the

surveillance CLIST. After the suspension, the CLIST is invoked again to check XRF takeover processing.

ALTSTART=YES|NO
This parameter is used for XRF systems only and is required if XRF=YES. When the control region is started and is the *active* in an XRF complex, this parameter specifies whether the *alternate* IMS control region should also be started immediately after the *active* becomes available.

ALTRST=YES|NO
This parameter is used for XRF systems only. After an XRF takeover, this parameter specifies whether a new *alternate* should be started to replace the old *active* IMS subsystem.

TKO={YES|NO}
This parameter is used for XRF systems only. If the XRF complex is defined so that a takeover will not automatically occur (specified in the DFSHSB*xx* member of the IMS PROCLIB), a message will be issued to the MVS system console so the operator can respond to allow the takeover to occur or to stop the process. If the TKO parameter specifies YES, then IMS Automation will respond with the affirmative reply to this message; see “DFS3869A/DFS3869I—Alternate XRF IMS subsystem detects potential failure” on page 66.

PRTNRID=*XRF partner subid*
This parameter is used for XRF systems only and is required if XRF=YES. This is the subsystem ID for the partner IMS control region. This control region may be specified either in the local (same) or a remote domain.

PRTNRDOM=*XRF or FDR partner domain*
This parameter is used for XRF or FDR enabled systems only. For an XRF environment specify this parameter and XRF=YES. For an FDR enabled system specify this parameter and XRF=NO. This is the NetView domain ID for the partner IMS control region in an XRF environment, or, the tracking FDR region in an FDR environment. This may be the local (same) or a remote domain.

PRTNRSUB=*XRF or FDR partner subsys*
This parameter is used for XRF or FDR enabled systems only. For a XRF environment specify this parameter and XRF=YES. For an FDR enabled system specify this parameter and XRF=NO.

For the XRF CONTROL region this is the SA OS/390 subsystem name of the alternate/backup control region. For the dependant regions this is the AOC/MVS subsystem name of the corresponding XRF partner dependant regions.

For an FDR enabled system, this is the SA OS/390 subsystem name for the FDR region.

WTCHNOTF=60|*time*
This parameter is used for XRF systems only and is required if XRF=YES. This parameter can be used to tailor the XRF takeover surveillance processing. It is specified in seconds.

If surveillance processing does not detect a change in the XRF status for the monitored IMS subsystem prior to the time specified in WTCHNOTF, additional IMS Automation warning messages will be issued. These messages describe possible problems with the takeover. Once issued, the clock starts again and the messages are not issued until the time spent in surveillance once again exceeds the time specified in WTCHNOTF.

WTCHALERT=60|*time*
This parameter is used for XRF systems only and is required if XRF=YES. This parameter can be used to tailor the XRF takeover surveillance processing. It is specified in seconds.

Once surveillance processing exceeds the time specified in this parameter with no change in the XRF status, additional messages are issued indicating that operator intervention may be required to complete the takeover. Once issued, the messages will be issued again every WDELAY time until an XRF status change is detected.

EXITS=(*user_exit***)**
This parameter specifies that a user exit will be called. This user exit is invoked only at shutdown of the IMS control region. The user exit must be a CLIST and a member of the DSICLD concatenation of NetView.

CTLNAME=*name*
name is the SA OS/390 subsystem name of the owning control region. This parameter is necessary only when a dependent region is defined as a dependent of more than one IMS control region. This parameter enables IMS Automation to determine how to display, start, and stop the region. The CTLNAME keyword is only valid with multiparent support offered by AOC/MVS Release 2 or later.

STRTTIMER = 5|*time*
This parameter is used for IMS initialization. This timer defines the interval for IMS Automation to wait before proceeding with IMS subsystem monitoring. IMS Automation will wait this interval before scheduling the function to report on the progress of the startup of an IMS complex. CLIST EVIEI00J is scheduled using this timer.

DBCTL = YES|NO
This parameter indicates whether the control region is a DB control region. (If YES, then **SUBTYPE=CTL** and **XRF=NO** must also be specified.)

Comments and Usage Notes

⇒ **Required**. ENVIRON is a required control file entry.

Examples of Usage

Example 1 Non-XRF Control Region

```
IMSIMSZ    ENVIRON,SHUTGO=YES,
           SUBID=IMSZ,
           XRF=NO,
           DEFSTART=AUTO,
           SUBTYPE=CTL
```

Example 2 XRF Control Region

```
IMSIAOCA   ENVIRON,SHUTGO=YES,
           SUBID=IMSW,
           XRF=YES,
           DEFSTART=AUTO,
           SUBTYPE=CTL,
           WDELAY=20,
           ITIMER=8,
           ALTSTART=YES,
           ALTRST=YES,
           TKO=YES,
           PRTNRID=IMSL,
           PRTNRDOM=AOCS6,
           PRTNRSUB=IMSIAOCB
```

Example 3 Non-XRF DBRC Region

```
DBRCIMSZ   ENVIRON,
           SUBTYPE=DBRC
```

Example 4 XRF DBRC Region

```
DBRIMSZ    ENVIRON,
           SUBTYPE=DBRC
```

Example 5 Non-XRF DLISAS Region

```
DBRCIAOCA  ENVIRON,
           SUBTYPE=DLS
```

Example 6 XRF DLISAS Region

```
DBRIMSZ    ENVIRON,
           SUBTYPE=DLS
```

Example 7 Non-XRF Message Region

```
MSGIAOCA   ENVIRON,
           SUBTYPE=TP
```

Example 8 XRF Message Region

```
MSGAOCA    ENVIRON,
           SYNCH=Y,
           SUBTYPE=TP,
           PRTNRSUB=MSGIMSLA
```

Example 9 Non-XRF and XRF Fast Path Region

```
FPIMSZA    ENVIRON,
           SYNCH=Y,
           SUBTYPE=FP,
           PRTNRSUB=FPIMSYA
```

Example 10 Batch Message Region, Non-XRF and XRF

```
BMPIMSA    ENVIRON,
           SYNCH=Y,
           SUBTYPE=BMP,
           PRTNRSUB=BMPIMSB
```

Example 11 Other Subsystem Region Child of IMS CTL Region

```
OTHERSYS   ENVIRON,
           SUBTYPE=OTHER
```

Example 12 DB Control Region

```
INSIMSJ    ENVIRON,
           SUBTYPE=CTL,
           SUBID=IMSJ,
           XRF=NO,
           DBCTL=YES,
           DEFSTART=AUTO
```

Example 13 FDR Enabled Control Region

```
INSIMSJ    ENVIRON,
           SUBID=IMSZ,
           XRF=NO,
           DEFSTART=AUTO,
           SUBTYPE=CTL,
           PRTNDOM=AOC10,
           PRTNRSUB=IMS10F
```

ENVIRON SETUP—Identify startup routine

This SA OS/390 ENVIRON SETUP, EXITS parameter identifies the IMS startup CLIST.

Keyword and Parameter Definitions

EVIEE001
EVIEE001 identifies the IMS startup CLIST.

Comments and Usage Notes

⇒ **Required**. You must code EVIEE001.

Refer to the SA OS/390 customization dialogs to build this control file entry.

Add EVIEE001 as an exit for the EXITS parameter of the SA OS/390 ENVIRON SETUP control file entry.

Examples of Usage

Example

```
ENVIRON    SETUP,  ...,
                   ...,
                   EXITS=(EVIEE001,USREXIT1,...)
```

EXTCOND—External conditions

Use this optional entry to define trigger names and definitions so that they can be controlled through the operator interface and to define those triggers that are unset in a status file subsystem record when certain startup and shutdown events occur.

This entry is not subsystem specific and applies to all IMS subsystems on this NetView domain.

```
EXTCOND event[,DESC=description]
             [,UNSET={START|UP|DOWN}]
```

Keyword and Parameter Definitions

event
The symbolic trigger name that represents an external event.

DESC=
Use this keyword to provide a description for the IMS Automation panel that lists the trigger events.

UNSET=
Use this keyword to unset a trigger in a status file subsystem record when the following conditions are met for that subsystem:

START A startup was initiated.
UP A startup is complete.
DOWN A shutdown is complete.

Comments and Usage Notes

1. The *event* name used here matches the *event* name as it is posted to the status file subsystem record. Refer to "IMSPOST—Post an external event" on page 172.
2. The *event* name used here matches the *event* name described in "TRIGGER—Startup and shutdown triggers" on page 142, where trigger requirements are defined for specific subsystems.
3. This control file entry is not subsystem specific. When using the UNSET keyword, verify that the trigger name (*event*) is only used on those subsystems that you want the trigger to be unset when the unset conditions are met.
4. SERVICE is a reserved word and cannot be used for a trigger name (*event*).
5. The CMD/REPLY control file entry can be used instead of the UNSET keyword to unset triggers for specific subsystems.

Examples of Usage

Example 1

```
EXTCOND PAYROLL,DESC='The payroll job is done'
EXTCOND ACCREC,DESC='Accounts receivable is done'
EXTCOND DAYEND,DESC='Day-end jobs are complete'
EXTCOND BATCH1,DESC='BATCH1 jobs are done'
EXTCOND BATCH2,DESC='BATCH2 jobs are done'
EXTCOND FORMAT,DESC='The OLDS database is formatted'
```

This example simply defines the trigger names and their descriptions. This appears on the panel that lists triggers for subsystems.

Example 2

```
EXTCOND FORMAT,DESC='The OLDS database is formatted',
        UNSET=START
```

This example specifies that when the trigger named FORMAT is set in a status file subsystem record, it is unset when a startup on that subsystem is initiated.

FORCE—Takeover by alternate XRF IMS subsystem

This entry defines the command to be issued when predatory takeover has occurred (the *alternate* IMS subsystem has taken control from the *active* IMS subsystem).

```
subsys  FORCE,CMD=('command')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

command
This is the command to be issued to initiate a forced shutdown.

Comments and Usage Notes

Required only for each XRF IMS subsystem.

This IMS command is issued when a predatory takeover has occurred (the *alternate* has assumed control). The command is issued on the old *active* to inform it that it is no longer the *active*.

One variable is available for use in the FORCE entry, &EHKVAR1.

Examples of Usage

Example

```
IMSIAOCA  FORCE,CMD=('MVS F &EHKVAR1,STOP')
```

&EHKVAR1 is the jobname. This is an MVS modify command that causes IMS to terminate immediately.

HOLDQ—Issue commands at shutdown

This entry defines the commands that may be issued at the initiation of the shutdown process and subsequent to PRECHKP.

```
subsys  HOLDQ,CMD=(,,'command')
            [,CMD=(,,'command')]
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

command
Any appropriate MVS/JES command may be coded.

Comments and Usage Notes

⇒ **Not required**.

These MVS commands are issued at the very beginning of the shutdown process. Any appropriate MVS/JES commands may be coded; however, the original intent was to specify JES commands that would **hold** certain job classes that BMPs run in. This would prevent other BMPs from starting once the shutdown process was initiated.

Examples of Usage

Example

```
IMSIMSZ  HOLDQ,CMD=(,,'MVS $HQ,C=123')
```

IMSCNTL—IMS/DC control region

The required IMSCNTL entry describes the IMS/DC control region.

```
IMSCNTL  subsys,AUTOOPS=autoops_name,
                APPLID={applid|(generic_applid,specific_applid)},
                [,APPLID1=IMS appl]
                [,APPLID2=IMS appl]
                [,MAJNODE1=IMS major node]
                [,MAJNODE2=IMS major node]
                [,OTHMAJ1=other major node]
                [,OTHMAJ2=other major node]
                [,DEFLTHSB=1|2]
                [,PARMS1=startup parms]
                [,PARMS2=startup parms]
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

AUTOOPS=*autoops_name*
This identifies the NetView Automated Operators that will control this IMS system and its dependents. The value must be defined in an AUTOOP statement. Operators may be assigned to one or more IMS subsystems.

APPLID=
APPLID identifies the receiver/sender name when the program-to-program interface is initiated and used. Use one of the following:

applid
Use this keyword if this subsystem is defined as non-XRF.

(generic_applid,specific_applid)
If this subsystem is defined as XRF-capable, then use *generic_applid* as the name given to both the active and alternate subsystem and *specific_applid* as the specific name for this subsystem.

APPLID1/2=*IMS appl*
If you want SA OS/390 automation to check that the IMS APPLID is **connectible** prior to starting IMS, specify a value here for the IMS APPLID. APPLID1 or APPLID2 identifies the VTAM name.

Use the specification APPLID1 for a non-XRF IMS subsystem. If the system is an XRF IMS, use both the APPLID1 and the APPLID2 specifications to define both APPLIDs that can be used by the IMS subsystem. APPLID1 will be checked when the HSBID is **1**, APPLID2 will be checked when the HSBID is **2**.

Note that for startup of an *alternate* IMS subsystem, if the APPLID is specified and is not **connectible**, then startup processing will not continue for the *alternate*.

MAJNODE1/2=*IMS appl*
If you want SA OS/390 automation to check that the IMS VTAM Major Node is active (and make it active if it is not) prior to starting the IMS control region,

then specify the correct major node here. The specification for an XRF IMS subsystem is similar to that for APPLIDs.

OTHMAJ1/2=*IMS appl*
If you want SA OS/390 automation to check that any other VTAM Major Node is active (and make it active if it is not), then specify the correct node name here. The specification for an XRF IMS subsystem is similar to that for APPLIDs.

DEFLTHSB=*1***|***2*
For an XRF IMS control region, the DEFLTHSB specification describes the preferred HSBID to be used when restarting an IMS system in situations where it does not matter which HSBID is used (in terms of IMS recovery).

PARMS1/2=*startup parms*
If you have additional parameters that should be appended to the startup command to start IMS, then specify them here. For a non-XRF IMS subsystem, specify your local parameters using the PARMS1 parameter. For an XRF IMS subsystem, the parameters used will be based on the current HSBID. For example, if HSBID=1, then the parameters coded on the PARMS1 parameter will be used.

Comments and Usage Notes

Since a DB control region does not use VTAM, the **APPLID**, **APPLID***x*, **MAJNODE***x*, **OTHMAJ***x*, and **DEFLTHSB** keywords are not required. The **AUTOOPS** keyword, however, is required.

Examples of Usage

Example 1 Non-XRF

```
IMSCNTL   IMSIMSZ,APPLID1=IMS10Z,
          MAJNODE1=IMSPROD1,
          PARMS1='.IMSAPPLA,MSGC=X',
          APPLID=IMS10Z,
          AUTOOPS=IMZOPER
```

Example 2 XRF

```
IMSCNTL   IMSIAOCA,APPLID1=IMSAOC1,
          APPLID2=IMSAOC2,
          MAJNODE1=IMSPROD1,
          MAJNODE2=IMSPROD2,
          DEFLTHSB=1,
          PARMS1='.IMSAPPLA,MSGC=X',
          PARMS2='.IMSAPPLB,MSGC=X',
          APPLID=(IMSXRF,IMSAOC1),
          AUTOOPS=IMSXOPER
```

Example 3 DBCTL

```
IMSCNTL    IMS01C,
           PARMS1="PARM1='AUTO=Y'",
           AUTOOPS=IMSOP99
```

IMS CRITMSGS—Display critical messages in the SDF

To save critical messages under SDF's Critical Messages Handler, you must add CODE=SAVE to the IMS CRITMSGS control file entry.

Note: This entry is required for the common routine, IMSFWM, to work. See "IMSFWM—Add messages to SDF" on page 170.

```
IMS CRITMSGS,CODE=(msgs,,,{SAVE|NOSAVE})
```

This entry determines which messages are displayed on the panels that list IMS messages, where *msgs* is:

- An asterisk (*) to indicate all messages
- A message prefix followed by an asterisk to indicate a group of messages
- A specific message

SAVE and NOSAVE are used to indicate whether or not you want the messages saved.

Comments and Usage Notes

1. Categories of messages are itemized on the SDF IMS Monitor panel. Placing the cursor on an IMS Monitor panel item and pressing ENTER displays the list of messages saved for that category.
2. This is not an SDF keyword, but is part of the SDF definitions used to support IMS automation.
3. Refer to "SDF support" on page 110.
4. Refer also to "Setting Up SDF for IMS Automation" on page 25.

Examples of Usage

Example

```
IMS CRITMSGS,CODE=(*,,,SAVE)
```

This example states that all IMS messages are saved so that they can be displayed on the panels that list messages.

IMSGROUP—Group of IMS subsystems

Use this control file entry to define a set of IMS subsystems that can be:

- Started or shut down from the operator interface as a group
- Sent broadcast messages to as a group.

```
IMSGROUP groupname,MEMBER=subsys,...,MEMBER=subsys
```

Keyword and Parameter Definitions

groupname
The one- to eight-character symbolic name which will represent this group of IMS subsystems.

MEMBER=
The symbolic name (*subsys*) by which this IMS subsystem is known to SA OS/390, as defined with the SUBSYSTEM control file entry.

Comments and Usage Notes

1. A group can consist of subsystems from more than one NetView domain or system. However, it is recommended that group definitions that contain subsystems from more than one domain only be defined at the focal point.
2. This entry is only used for operator interface startup and shutdown and broadcast processes.
3. The requested procedure starts with the first subsystem MEMBER defined in the entry and steps through each MEMBER one at a time.
4. Service periods, triggers, and XRF information are checked before a subsystem startup or shutdown process is initiated. If a subsystem does not meet the required conditions, the operator is asked whether or not to proceed. After the operator responds, the requested process continues with either the current subsystem or with the next subsystem listed.

Examples of Usage

Example

```
IMSGROUP IMSGRP,MEMBER=IMS10AA,MEMBER=IMS10AB,MEMBER=IMS10Z
```

In this example, the requested procedure starts with IMS10AA and steps through the next two subsystems one at a time.

IMSRSENM—IMS resource name

The IMSRSENM entry should be coded as follows for each XRF IMS complex defined, to match the XRF RSENAME to the IMS subsystems with which it is associated.

IMSRSENM is read by the command lists EVIEO007 and EVIEO009.

```
IMSRSENM  rseid,SUBSYSTEM1=subsys1, SUBSYSTEM2=subsys2
```

Keyword and Parameter Definitions

rseid
This is the value specified as the RSENAME for the XRF complex as defined in the DFSHSB*xx* member in the IMS PROCLIB.

subsys1
This parameter is the SA OS/390 subsystem name of one of the IMS control region subsystems that participate in an XRF complex.

The order in which the IMS subsystems are specified does not matter. The SUBSYSTEM1 and SUBSYSTEM2 keywords are not related to HSBID.

subsys2
This parameter is the partner IMS control region subsystem name that participates in this XRF complex.

Comments and Usage Notes

Required only for an XRF IMS subsystem.

If both of the IMS control regions in the XRF complex are defined in the same SA OS/390 control file member, then this IMSRSENM entry should only be defined once. If the IMS control regions are defined in separate SA OS/390 control file members, then each control file member must have an identical IMSRSENM entry.

Examples of Usage

Example

```
IMSRSENM   XRFA,SUBSYSTEM1=IMSIAOCA,SUBSYSTEM2=IMSAOCB
```

INITSTART—Control subsystem start

The INITSTART flag entry provides an “on/off” switch to control the initial start of an IMS subsystem.

```
INITSTART subsys,AUTO=E,
                 EXITS=(EVIEIEXT{,userexit...,userexit})
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

AUTO=E
Indicates that the automation control exits, defined by the EXITS parameter, will be called to determine whether automation should continue.

EXITS=
Defines the automation control exits to be called.

EVIEIEXT
Identifies the exits parameter required for the initial startup of an IMS subsystem. Code exactly as shown.

Comments and Usage Notes

⇒ **Required**.

Refer to the SA OS/390 customization dialogs to build this control file entry.

This option allows control over whether SA OS/390 will automatically initiate an IMS startup. For example, you must specify AUTO=E if you want the SETSTATE command with the START option to cause the IMS startup to be initiated.

The EVIEIEXT exit can still be used if AUTO=N is set. This allows for IMS to be started only through the operator interface. If EVIEIEXT is the last exit in the list it issues return code ‘100’ instead of ‘0’ indicating that the automation is off but that the startup still proceeds. Otherwise it returns ‘0’ to hold the customer’s exit responsible for the startup. Any return code other than 0 from the customer’s exit stops the startup, for example:

```
/*REXX*/
Arg argstring
Parse Var argstring flag autoind subsapp1 substype .
If autoind = 'NOAUTO' then
  return_code = 99      /* anything but 0 */
Else
  return_code = 0
Return return_code
```

Examples of Usage

Example

```
INITSTART  IMS10AA,AUTO=E,EXITS=(EVIEIEXT)
```

IPS—XRF IMS takeover commands

This entry defines the commands that may be issued during the XRF IMS takeover process.

```
subsys  IPS,CMD=(,,'command')
          [,CMD=(,,'command')]
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

command
Any appropriate MVS command may be coded.

Comments and Usage Notes

⇒ **Not required**. Applicable to XRF IMS environment only.

These MVS commands are issued during an XRF IMS takeover process on the system where the new *active* IMS subsystem resides. Due to the change in workload characteristics resulting from an XRF IMS takeover, it may be advantageous to code optional MVS commands to make adjustments to MVS processing.

Examples of Usage

Example

```
IMSIAOCA   IPS,CMD=(,,'MVS SET IPS=02')
```

NTFYOP—Define notification operator

NTFYOP identifies the people who receive notifications. NTFYOP is an optional control file entry, but customizing these entries for IMS Automation is recommended.

```
NTFYOP operid[,CLASS={60|(class,60,...)}]
```

Keyword and Parameter Definitions

CLASS Add the class “60” to the NTFYOP entry for all NetView operators you want to receive IMS automation notifications.

Comments and Usage Notes

Refer to the SA OS/390 customization dialogs to build this control file entry.

Examples of Usage

Example

This example defines an operator to receive IMS Automation notifications only.

```
NTFYOP NETOP1,CLASS=(60)
```

In this example, NETOP1 is defined to receive IMS Automation notifications.

OLDS—Define recovery criteria for OLDS

The OLDS control file entry defines automation settings for online data sets (OLDS). IMS Automation monitors the OLDS at regular intervals and notifies operators before the IMS fails. Using the OLDS entry, you specify the minimum number of OLDS that must be available at all times. If the number of available OLDS drops below that minimum, IMS Automation starts the OLDS you designate as spares. If the number of available OLDS exceeds the minimum, IMS Automation will stop unneeded OLDS. Please read the usage notes carefully.

During normal IMS operations, the archive jobs copy data from the OLDS to the system log data sets (SLDS) to make the OLDS available for more data. When archive jobs fail, the OLDS fill up and are no longer available. IMS Automation monitors how long it takes the archive jobs to execute and notifies operators if the archive jobs are taking longer than expected to run. Additionally, IMS Automation checks that the number of OLDS data sets with an 'OTHER-STS' of 'BACKOUT' has not exceeded the value specified in the BACKOUT operand. By monitoring the archive jobs, IMS Automation helps operators detect possible problems early so they can be corrected before the OLDS fill and IMS stops.

Attention: Investigate any problems with OLDS immediately and correct. Failure to do so can cause IMS to stop.

```
subsys OLDS[,MINIMUM=nn]
           [,SPARES=(nn,nn...)]
           [,ARCHIVETIME=hh:mm:ss]
           [,RETRYCNT=n]
           [,BACKOUT=nn]
```

Keyword and Parameter Definitions

subsys
The name of the IMS.

MINIMUM=*nn*
The minimum number of OLDS that must be available at all times. The default minimum is 50% of the *normal number* of OLDS. The normal number of OLDS display when you issue the DISPLAY OLDS command (DIS OLDS). The normal number can not be less than three and includes all OLDS that IMS started at initialization or that operators started. OLDS that are defined but are not listed in the response to the DISPLAY OLDS command are not included in calculating the normal number of OLDS.

SPARES=**(***nn*,*nn*...**)**
The spares are OLDS that IMS Automation activates when the number of available OLDS drops below the minimum. The names for the spares are the two-digit numbers taken from the end of the ddname. For example, DFSOLP99 is the spare named 99. Be sure that the names of the spares match the names of existing OLDS.

ARCHIVETIME=*hh:mm:ss*
The archive time is the maximum length of time archive jobs take to run. Set this timer to be longer than the archive jobs take to execute. The default setting is 00:10:00 (10 minutes).

RETRYCNT=*n*
The retry count is the number of times that IMS Automation will attempt to get an outstanding reply ID when activating or deactivating a spare OLDS. The default is 5 retries.

BACKOUT=*nn*
The maximum number of OLDS that can have an OTHER-STS of BACKOUT. Set this number to match the total number of acceptable OLDS data sets with an OTHER-STS of BACKOUT.

Comments and Usage Notes

1. Make sure that the OLDS you designate as spares exist.

   Note: To define spares as dynamically allocated additional OLDS, preallocate and catalog candidate data sets and specify data set names using the dynamic allocation macro, DFSMDA. Provide DFSMDA members for all OLDS. For information on using DFSMDA, refer to the *IMS/ESA Utilities Reference*.

2. Do not include the spare OLDS in DFSVSM00, which is the member that tells IMS which data sets to start when IMS initializes. IMS Automation will stop any spare OLDS and remove them from SDF if the minimum number of OLDS are available.

3. If you activate a spare OLDS manually while the number of available OLDS is equal to or above the minimum, IMS Automation removes the spare from IMS and deletes any SDF entries for the spare.

   Note: IMS Automation does ***not*** send notifications when it stops an OLDS that is designated for use only as a spare.

4. If you define new OLDS, you must stop and restart IMS to use these settings.

5. For IMS Automation to monitor OLDS, you must define an OLDS control file entry.

6. The value of OLDS BACKOUT can not exceed the total number of OLDS data sets defined to the IMS subsystem. The value should represent an acceptable maximum number of OLDS data sets with an OTHER-STS of BACKOUT.

Examples of Usage

```
IMS10AA OLDS,
          MINIMUM=5,
          SPARES=(97,98,99),
          ARCHIVETIME=00:20:00,
          RETRYCNT=7
          BACKOUT=2
```

For this example, the minimum number of OLDS that must be kept available at all times is five. If the number of available OLDS is less than five, IMS Automation starts one of the spares (DFSOLP97, DFSOLP98, or DFSOLP99). If IMS Automation does not receive an outstanding reply ID within seven tries, no further attempts are made. IMS Automation also sends notifications to operators if the archive jobs take longer than 20 minutes to execute or the total number of OLDS data sets with an OTHER-STS of BACKOUT exceeds a maximum of 2.

POSTCHKP—Issue commands after IMS shutdown checkpoint

This entry defines the commands that may be issued immediately after the IMS shutdown checkpoint has been issued.

```
subsys  POSTCHKP[,CMD=(,,'command')]
                [,CMD=(,,'command')]
                [,REPLY=(,RETRYn,'ims_command')]
                [,REPLY=(,RETRYn,'ims_command')]
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

command
Any appropriate command may be coded.

ims_command
Any appropriate IMS command may be coded.

Comments and Usage Notes

⇒ **Not required**.

For DB control regions, use the **CMD=** syntax only.

Attention: Although replies are supported, their use at this point in the processing of an IMS is not recommended. Because POSTCHKP occurs after the checkpoint call has been issued, it is likely that no outstanding reply is available. Improper use of the reply function with POSTCHKP can cause performance problems.

These commands (CMD=) and IMS replies (REPLY=) are issued just after the IMS shutdown checkpoint has been issued. There may be multiple commands and IMS replies.

Two variables are available for use in the commands:

&EHKVAR1 This is the value of the VTAM IMS APPLID.

&EHKVAR2 This is the value of the subsystem ID (*subid*) specified in the SUBID parameter in the IMS ENVIRON entry.

Examples of Usage

Example

```
IMSIMSZ  POSTCHKP,CMD=(,,'MVS $AQ,C=123'),
                CMD=(,,'MVS S IMSJOB'),
                REPLY=(,RETRY5,'/ASSIGN LTERM MASTER LINE 1 PTERM 1'),
                REPLY=(,RETRY5,'/IDLE NODE ALL')
```

PRECHKP—Issue commands prior to IMS shutdown checkpoint

This entry defines the commands that may be issued immediately prior to the IMS shutdown checkpoint being issued but prior to HOLDQ.

```
subsys  PRECHKP[,CMD=(,,'command')]
               [,CMD=(,,'command')]
               [,REPLY=(,RETRYn,'ims_command')]
               [,REPLY=(,RETRYn,'ims_command')]
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

command
Any appropriate command may be coded.

ims_command
Any appropriate IMS command may be coded.

Comments and Usage Notes

⇒ **Not required**.

For DB control regions, use the **CMD=** syntax only.

These commands (CMD=) and IMS replies (REPLY=) are issued just prior to the IMS shutdown checkpoint being issued. There may be multiple commands and IMS replies.

Examples of Usage

Example

```
IMSIMSZ  PRECHKP,CMD=(,,'MVS $HQ,123'),
                 CMD=(,,'MVS S IMSJOB'),
                 REPLY=(,RETYR5,'/STO TRAN ALL'),
                 REPLY=(,RETYR5,'/STO NODE ALL')
```

PRODUCT—Subsystem state/action table sets

Use this entry to identify a set of state/action tables to be used for each IMS subsystem.

```
PRODUCT product,SUBSYS=(subsys,AREA=area)
               [,SUBSYS=(subsys,AREA=area)]
               .
               .
               .
               [,SUBSYS=(subsys,AREA=area)]
```

Keyword and Parameter Definitions

product
: The product for which these state/action tables are being defined. In this case, it will be IMS. If we were automating CICS and intended to use state/action tables, then the product would be CICS.

subsys
: The symbolic name by which this IMS subsystem is known to SA OS/390, as defined with the SUBSYSTEM control file entry.

AREA=
: Identifies the set of state/action tables to be used for this subsystem. This points to the control file entry described in “AREA—Define a set of state/action tables” on page 52.

Comments and Usage Notes

1. If a subsystem is not defined with this keyword, state/action tables are not used.
2. There must also be an AREA entry defined.
3. Refer to “Using State/Action Tables” on page 36 for more information.

Examples of Usage

Example

```
PRODUCT IMS,SUBSYS=(IMS10AA,AREA=IMS1)
```

Indicates that the set of state/action tables known as IMS1, as defined with the AREA control file entry, are used for this subsystem, IMS10AA.

RECONS—Set monitoring interval for RECONS

The RECONS control file entry turns on active monitoring at regular intervals for RECONS. RECONS are critical for IMS recovery control. At the interval specified on the RECONS entry, IMS Automation checks to make sure that a spare RECON is available. If no spare is available, IMS Automation sends a notification to the operators.

IMS Automation also does the following RECONS automation:

- Informs operators when there has been a switch from one RECON to another
- Checks for exception conditions, such as I/O errors

For these automation actions, which occur in response to error messages, you do not need to code any control file settings.

```
subsys RECONS,MONITOR=hh:mm:ss
```

Keyword and Parameter Definitions

subsys
The name of the IMS.

MONITOR=*hh:mm:ss.*
The setting on the MONITOR keyword determines how often IMS Automation checks the RECONS to make sure a spare is available.

Comments and Usage Notes

1. IMS Automation does some RECONS checking even if no RECONS control file entry is defined:
   - IMS Automation checks RECONS at IMS initialization by invoking the EVIECR04 module in the automation table.
   - IMS Automation notifies operators when there is a switch from one RECON to another.
2. For IMS Automation to actively monitor the RECONS for spares, you must specify a monitoring interval on the RECONS control file entry.

Examples of Usage

```
IMS10AA  RECONS,
              MONITOR=00:15:00
```

In this example, IMS Automation checks the RECONS for spares every 15 minutes.

RECOVERY—Turn automation on/off

The extended recovery flag entry provides the capability to turn automation on or off for an entire IMS or for its minor resources. For the minor resources, you can either specify the minor resources as a generic group or you can specify recovery for a single, specific minor resource.

For example, if you code the generic minor resource, MSC, recovery is set for all the MSCs on the subsystem. If add the link ID (MSC.*link_id*), then recovery is set only for that one MSC link.

```
RECOVERY subsys[.minor_resource],AUTO=Y|N
```

Keyword and Parameter Definitions

subsys
The name of the IMS subsystem as defined on the SA OS/390 SUBSYSTEM control file entry.

.minor_resource
IMS Automation supports the following minor resources (also known as extended automation flags) on the RECOVERY entry:

MSC	All Multiple Systems Couplers (MSC) links on the IMS
MSC.*link_id*	A single MSC link
OLDS	All online data sets (OLDS)
PROG	All program-driven BMP failures
PROG.*prog_id*	A single program-driven BMP failure
TRAN	All transaction failures
TRAN.*trans_id*	A single transaction failure

Comments and Usage Notes

Refer to the SA OS/390 customization dialogs to build this control file entry.

Although only the AUTO=Y|N keyword is documented here, the RECOVERY control file entry supports all other SA OS/390 keywords including:

EXITS=
NOAUTO=
ASSIST=

You can set the parameters for the RECOVERY control file entries through the SA OS/390 customization dialogs. If you want to code these keywords manually using an editor, browse the RECOVERY entries in the SA OS/390 control file for examples of usage. Exits can be used if AUTO=N is set. For further information, refer to “Comments and Usage Notes” in the section on “INITSTART—Control subsystem start” in this chapter.

Examples of Usage

Example 1 subsys.MSC.link_id

```
RECOVERY IMS10AA.MSC.0001,AUTO=Y
```

0001 is a specific link number, so this entry is valid for this single MSC only.

Example 2 subsys.TRAN

```
RECOVERY IMS10AA.TRAN,AUTO=N
```

This entry is used by all transactions except those transactions that have their own individual entries.

Example 3 subsys.TRAN.trans_id

```
RECOVERY IMS10AA.TRAN.SAMPLE1,AUTO=Y
```

SAMPLE1 is the transaction ID.

Example 4 subsys.PROG

```
RECOVERY IMS10AA.PROG,AUTO=Y
```

Example 5 subsys.PROG.prog_id

```
RECOVERY IMS10AA.PROG.SAMPLE2,AUTO=N
```

SAMPLE2 is the program ID. This entry applies only to the program-driven BMP named SAMPLE2.

Example 6 subsys.OLDS

```
RECOVERY IMS10AA.OLDS,AUTO=Y
```

RELEASEQ—Issue commands after shutdown completes

This entry defines the commands that may be issued after IMS shutdown completion.

```
subsys  RELEASEQ,CMD=(,,'command')
                [,CMD=(,,'command')]
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

command
Any appropriate MVS command may be coded.

Comments and Usage Notes

⇒ **Not required**.

These MVS commands are issued after an IMS control region has completed shutdown. Any appropriate MVS/JES commands may be coded, however, the original intent was to perform the reverse function of the HOLDQ commands (refer to "HOLDQ—Issue commands at shutdown" on page 87), that is, to release the job classes held during the shutdown process.

Examples of Usage

Example

```
IMSIMSZ  RELEASEQ,CMD=(,,'MVS $AQ,C=123')
```

RESTART—Control restart of IMS subsystem

The RESTART flag entry provides an “on/off” switch to control the restart of an IMS subsystem.

```
RESTART subsys,AUTO=E,
               EXITS=(EVIEIEXT{,userexit...,userexit})
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

AUTO=E
Indicates that the automation control exits, defined by the EXITS parameter, will be called to determine whether automation should continue.

EXITS=
Defines the automation control exits to be called.

EVIEIEXT
Identifies the exits parameter required for the restart of an IMS subsystem. Code exactly as shown.

Comments and Usage Notes

⇒ **Required**.

Refer to the SA OS/390 customization dialogs to build this control file entry.

This entry must be specified as shown. If desired, additional exits may be specified on the EXITS parameter, however, EVIEIEXT must be **first** in the list. Exits can be used if AUTO=N is set. For further information, refer to “Comments and Usage Notes” in the section on “INITSTART—Control subsystem start” in this chapter.

Examples of Usage

Example

```
RESTART  IMS10AA,AUTO=E,EXITS=(EVIEIEXT)
```

RESTARTABORT—Emergency restart commands

This entry defines the emergency restart commands required (for response to the DFS166 and DFS0618A messages) following an abort termination of the restart process.

```
IMS Control Region Syntax:
subsys  RESTARTABORT,REPLY=(OVERRIDE,RETRYn,'command'),
                     REPLY=(BACKUP,RETRYn,'command'),
                     REPLY=(ERE,RETRYn,'command')

DB Control Region Syntax:
subsys  RESTARTABORT,CMD=(OVERRIDE,,'command'),
                     CMD=(ERE,,'command')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

OVERRIDE
When message DFS618A is issued, the OVERRIDE reply will be issued for a non-XRF IMS subsystem, or an XRF IMS *active* subsystem when its partner subsystem is not currently active.

BACKUP
When message DFS618A is issued, the BACKUP reply will be issued for an XRF IMS alternate subsystem when its partner is currently active.

ERE
When message DFS166 is issued for an IMS subsystem and a valid checkpoint was not found during a restart, the ERE reply will be issued.

command
Code the *command* for OVERRIDE, BACKUP, and ERE replies as specified in the example below.

Comments and Usage Notes

This reply is issued in response to the RESTARTABORT for either the DFS0618A or DFS166 message.

Examples of Usage

Example 1 IMS Control Region

```
IMSIMSZ    RESTARTABORT,REPLY=(OVERRIDE,RETRY5,'/ERE OVERRIDE.'),
                        REPLY=(BACKUP,RETRY5,'/ERE BACKUP.'),
                        REPLY=(ERE,RETRY5,'/ERE.')
```

Example 2 DB Control Region

```
IMS01C     RESTARTABORT,
             CMD=(OVERRIDE,,'MVS &EHKVAR7ERE COLDSYS OVERRIDE.'),
             CMD=(ERE,,'MVS &EHKVAR7ERE.'),
```

SDF support

Several words were added to support the SDF IMS Monitor panel. The words are coded in the same way that statuses or resources are coded in SA OS/390. These words are used to determine the:

- Priority that determines color propagation
- Color override

IMSARCH IMS online log archive
IMSMSCL IMS MSC links
IMSOLDS IMS OLDS
IMSRECN IMS RECON
IMSTRAN IMS transactions
IMSTIMR IMS timers
| **IMSSTRCT** IMS CF structures
CRITMSG The default critical messages definition (IMS Critical Message)
CRITMSGA Messages ending in A (IMS Critical Message)
CRITMSGE Messages ending in E (IMS Critical Message)
CRITMSGW Messages ending in W (IMS Critical Message)
CRITMSGI Messages ending in I (IMS Critical Message)

Comments and Usage Notes

1. SDF must be coded in an %INCLUDE file and edited. There is no SA OS/390 customization dialog that supports this control file entry.
2. The PRIORITY and COLOR keywords are described in the SA OS/390 SDF programming information.
3. Placing the cursor on a IMS Monitor panel entry and pressing ENTER displays the list of messages saved for that category.
4. The IMS CRITMSGS,CODE=(*,,,SAVE) entry shown in the example states that all messages are saved so that they can be displayed on the panels that list messages. You could use this entry to specify messages that you don't want saved by using the NOSAVE option, as described in "IMS CRITMSGS—Display critical messages in the SDF" on page 90. This is not an SDF keyword, but is part of the SDF definitions used to support IMS automation.
5. Refer to "Setting Up SDF for IMS Automation" on page 25.
6. It is recommended that you use the default priority and color definitions.

Examples of Usage

The SA OS/390 control file entries like this:

```
IMS      CRITMSGS,CODE=(*,,,SAVE)
SDF      IMSARCH,PR=200,CO=R
SDF      IMSMSCL,PR=200,CO=R
SDF      IMSOLDS,PR=200,CO=R
SDF      IMSRECN,PR=200,CO=R
SDF      IMSTIMR,PR=200,CO=R
SDF      IMSTRAN,PR=200,CO=R
SDF      IMSSTRCT,PR=200,CO=R,CLEAR=(Y,RV)
SDF      CRITMSG,PR=503,CO=G
SDF      CRITMSGA,PR=500,CO=R
SDF      CRITMSGE,PR=501,CO=Y
SDF      CRITMSGW,PR=502,CO=T
SDF      CRITMSGI,PR=503,CO=G
```

When coded this way, the first entry states that all messages will be saved to be listed on subsequent panels. The next seven entries specify that the IMS Monitor panel item associated with the entry turns red when a message is logged for that category. The CRITMSG, CRITMSGA, CRITMSGE, CRITMSGW, and CRITMSGI keywords indicate what color the IMS Critical Message entry turns based on the types of messages logged. Because CRITMSGA has the highest priority in this group (the lower the number, the higher the priority), this color will override any of the other color definitions if a message ending in A is logged.

SERVICE—Service period definitions

Use this entry to define the service periods in which this IMS subsystem is to be active. This initiates startup and shutdown processes.

```
SERVICE subsys,DAY=(dayx,nnnn-nnnn)
           [,DAY=(dayx,nnnn-nnnn...nnnn-nnnn)]
           .
           .
           .
           [,DAY=(dayx,nnnn-nnnn...nnnn-nnnn)]
```

Keyword and Parameter Definitions

subsys
The symbolic name by which this IMS subsystem is known to SA OS/390, as defined with the SUBSYSTEM control file entry.

DAY=
Use this parameter to define the days and times that this IMS subsystem is active. The variables *dayx* can be expressed as:

- DAILY (every day of the week)
- WEEKEND (Saturday and Sunday)
- WEEKDAY (Monday through Friday)
- MONDAY
- TUESDAY
- WEDNESDAY
- THURSDAY
- FRIDAY
- SATURDAY
- SUNDAY

The variable *nnnn-nnnn* represents the service period window. This is expressed in military time. For example, 0100-1900 indicates that this IMS subsystem will be active between 1:00 AM until 7:00 PM. Up to five service period windows can be defined for each DAY= entry. DOWN indicates that the subsystem is down for that day (from midnight to midnight).

Comments and Usage Notes

1. The four asterisks (****) indicate that the subsystem remains active into the next day. Thus, if one day ends with four asterisks, the next day must begin with four asterisks.
2. DOWN can only be used as a single service period for a given day specification. The format must be DOWN-DOWN (it begins and ends with DOWN).
3. Specific days override DAILY, WEEKEND, and WEEKDAY.
4. WEEKEND and WEEKDAY override DAILY.

Examples of Usage

Example 1

```
SERVICE IMS10AA,
     DAY=(WEEKDAY,0830-1700)
```

In this example, IMS10AA is up from 8:30 in the morning until 5:00 in the afternoon during the week.

Example 2

```
SERVICE IMS10AA,
   DAY=(WEEKDAY,0830-1700)
   DAY=(WEEKEND,DOWN-DOWN)
```

Example 1 and example 2 are the same, because DOWN is the default.

Example 3

```
SERVICE IMS10AA
   DAY=(DAILY,****-0200,0400-****)
```

This example shows a daily shutdown between 2 a.m. and 4 a.m. The service period will continue across midnight.

SHUTINIT—Initial shutdown commands

This entry defines the initial shutdown commands for the subsystem.

```
subsys SHUTINIT,
       CMD=(,,'PUT AN INITIAL SHUTDOWN COMMAND IN PLACE')
```

Keyword and Parameter Definitions

subsys
The SA OS/390 subsystem name.

Comments and Usage Notes

Refer to the SA OS/390 customization dialogs to build this control file entry.

The parameter:

- is optional.
- can be coded for the IMS control region, message region, fast path region, and the batch message region.

Examples of Usage

Example for IMS Message Region without SHUTOPTIONS=PARENT

```
MSGIMSZA     SHUTINIT,
             CMD=(,,'MSG ALL MSGIMSZA IS COMING DOWN')
```

SHUTNORM/SHUTIMMED/SHUTFORCE—Control region

These three control file entries are specified to support operator selection of the "NORMAL, IMMEDIATE, or FORCE" shutdown type on the Operator Shutdown Panel for the IMS control region.

```
subsys  SHUTNORM,CMD=(PASS1,,'EVIET001 subsappl,NORM,shutoption')
subsys  SHUTIMMED,CMD=(PASS1,,'EVIET001 subsappl,IMMED,shutoption')
subsys  SHUTFORCE,CMD=(PASS1,,'EVIET001 subsappl,FORCE,shutoption')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

PASS1
Code exactly as shown. Unlike the SA OS/390 shutdown logic, a second pass will not be attempted for these entries. The logic in the EVIET001 CLIST will handle all subsequent shutdown activities.

EVIET001
Code exactly as shown. EVIET001 is the required shutdown CLIST.

subsappl
The subsystem name as specified in the control file.

shuttype **{NORM|IMMED|FORCE}**
A brief definition of the three *shuttypes* follows:

NORM causes a checkpoint to be issued and an attempt to shut down IMS in an orderly, structured manner; cancellation of message regions and the control region occurs after a predetermined time delay.

IMMED causes a checkpoint to be issued and immediate cancellation of message regions; cancellation of the control region occurs after a predetermined time delay.

FORCE causes immediate flushing of the entire IMS complex, including message and control regions.

Generally, the NORM parameter is specified with the SHUTNORM entry and the IMMED parameter is specified with the SHUTIMMED entry. Although it is possible, but not recommended, the NORM or IMMED parameters may be intermixed with either the SHUTNORM or SHUTIMMED entries (operator selection of the "Normal" or "Immediate" shutdown type from the Operator Shutdown panel relates to the SHUTNORM and SHUTIMMED entries respectively, however, the actual shutdown type is then dictated by either the NORM or IMMED parameter specified for each of these entries).

The FORCE parameter **must** be specified with the SHUTFORCE entry.

shutoption
A SHUTNORM and a SHUTIMMED entry requires one of the following shutdown options: DUMPQ, BACKUP, FREEZE, or PURGE. A SHUTFORCE entry requires either DUMP or NODUMP to be specified. DUMPQ and BACKUP are not applicable to DB control regions.

Comments and Usage Notes

Refer to the SA OS/390 customization dialogs to build this control file entry.

One of these commands is issued when a shutdown type of “Normal, Immediate, or Force” is selected from the Operator Shutdown panel. The SHUTNORM, SHUTIMMED, and SHUTFORCE entries work with the SHUTTYPES entry, related by their *shutoption* parameters (refer to “SHUTTYPES—Issue commands at operator shutdown” on page 120).

Examples of Usage

Example IMS Control Region

```
IMSIMSZ  SHUTNORM,
         CMD=(PASS1,,'EVIET001 IMSZMSZ,NORM,DUMPQ')
IMSIMSZ  SHUTIMMED,
         CMD=(PASS1,,'EVIET001 IMSZMSZ,IMMED,FREEZE')
IMSIMSZ  SHUTFORCE,
         CMD=(PASS1,,'EVIET001 IMSZIMSZ,FORCE,NODUMP')
```

SHUTNORM/SHUTIMMED/SHUTFORCE—Message, FP, and BMP regions

For regions defined without SHUTOPTIONS=PARENT, these entries define the shutdown commands to be used, based on the selection for IMS control region shutdown. If SHUTOPTIONS=PARENT is coded, these entries are not needed.

For BMPs that have their own outstanding replies, code the exact shutdown command used to terminate this region. SA OS/390 terminates the region using this command.

```
subsys   SHUTNORM,CMD=(PASS1,,'EVIET00J &SUBSAPPL,shuttype,')
subsys   SHUTIMMED,CMD=(PASS1,,'EVIET00J &SUBSAPPL,shuttype,')
subsys   SHUTFORCE,CMD=(PASS1,,'EVIET00J &SUBSAPPL,shuttype,')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS message, fast path, or BMP region.

PASS
SA OS/390 shutdown pass number.

EVIET00J
Code exactly as shown. EVIET00J is the required shutdown CLIST.

&SUBSAPPL
The subsystem name as specified in the control file.

shuttype
The *shuttype* parameter determines the type of shutdown (**NORM**, **IMMED**, or **FORCE**). The shutdown command that is issued for a given shutdown type depends on the type of region (message, fastpath, or BMP). The specific shutdown commands that are issued are coded on the control region's STOPREGION (page 130), STOPBMPREGION (page 128), or STOPFPREGION (page 129) control file entries. Table 7 on page 118 shows the relationship between the the region type, the *shuttype*, and these control file entries.

Table 7. Shutdown Commands for Message, Fastpath, and BMP Regions

Region type	Shutdown type (*shuttype*)	Control file entry where the shutdown command is coded
Message regions	NORM	*control_region* STOPREGION REPLY=(NORMAL,...)
	IMMED	*control_region* STOPREGION REPLY=(ABEND,...)
	FORCE	*control_region* STOPREGION REPLY=(CANCEL,...)
Fastpath regions	NORM	*control_region* STOPFPREGION REPLY=(NORMAL,...)
	IMMED	*control_region* STOPFPREGION REPLY=(ABEND,...)
	FORCE	*control_region* STOPFPREGION REPLY=(CANCEL,...)
BMP regions	NORM	*control_region* STOPBMPREGION REPLY=(NORMAL,...)
	IMMED	*control_region* STOPBMPREGION REPLY=(ABEND,...)
	FORCE	*control_region* STOPBMPREGION REPLY=(CANCEL,...)

Generally, the NORM parameter is specified with the SHUTNORM entry, the IMMED parameter is specified with the SHUTIMMED entry, and the FORCE parameter is specified with the SHUTFORCE entry.

When using these entries, the CLIST EVIET00J determines which shutdown command to issue by first determining whether this is a NORMAL, IMMEDIATE, or FORCE shutdown, and then associating that shutdown type with the type of message region (based on SUBTYPE entry from control file definitions).

- If this is an online message region, then the commands defined by the control file entry STOPREGION will be used. These commands were defined at the control region level.
- If this is a fast path message region, then the commands defined by the control file entry STOPFPREGION will be used. These commands were defined at the control region level.
- If this is a batch message region, then the commands defined by the control file entry STOPBMPREGION will be used. These commands were defined at the control region level. If this BMP region has specific shutdown commands then these must be coded, an example follows.

The FORCE parameter **must** be specified with the SHUTFORCE entry.

Examples of Usage

Example 1 IMS Message Region without SHUTOPTIONS=PARENT

```
MSGIMSZA SHUTNORM,
         CMD=(PASS1,,'EVIET00J &SUBSAPPL,NORM')
MSGIMSZA SHUTIMMED,
         CMD=(PASS1,,'EVIET00J &SUBSAPPL,IMMED')
MSGIMSZA SHUTFORCE,
         CMD=(PASS1,,'EVIET00J &SUBSAPPL,FORCE')
```

Example 2   IMS Fast Path Region

```
FPIMSZA  SHUTNORM,
         CMD=(PASS1,,'EVIET00J &SUBSAPPL,NORM')
FPIMSZA  SHUTIMMED,
         CMD=(PASS1,,'EVIET00J &SUBSAPPL,IMMED')
FPIMSZA  SHUTFORCE,
         CMD=(PASS1,,'EVIET00J &SUBSAPPL,FORCE')
```

Example 3   IMS Batch Message Region

```
BMPIMSA  SHUTNORM,
         CMD=(PASS1,,'USER-DEFINED SHUTDOWN COMMAND')
BMPIMSA  SHUTIMMED,
         CMD=(PASS1,,'EVIET00J &SUBSAPPL,IMMED')
BMPIMSA  SHUTFORCE,
         CMD=(PASS1,,'EVIET00J &SUBSAPPL,FORCE')
```

SHUTTYPES—Issue commands at operator shutdown

This entry specifies a command to be issued when an operator selects a shutdown type and (optional) shutdown option on the Operator Shutdown panel.

```
subsys SHUTTYPES,REPLY=(shutoption,RETRYn,'command'),
                 .
                 .
                 .
                 REPLY=(shutoption,RETRYn,'command'),
                 CMD=(NODUMP,,'command'),
                 CMD=(DUMP,,'command')
                 CMD=(PURGE,,'command')
                 CMD=(FREEZE,,'command')
```

Keyword and Parameter Definitions

subsys
: This is the SA OS/390 subsystem name of the IMS control region.

shutoption
: One SHUTTYPES entry must be specified, with four REPLY parameters (each specifying a different *shutoption* sub-parameter: DUMPQ, BACKUP, FREEZE or PURGE), and two CMD parameters (one specifying the DUMPQ *shutoption* sub-parameter, the other specifying the NODUMP *shutoption* sub-parameter).

'command'
: The IMS command to be issued, related to the *shutoption* sub-parameter specified in the REPLY or CMD parameter of the SHUTTYPES entry. The variable &EHKVAR1 may be used in the command; its value will be the IMS control region jobname.

Comments and Usage Notes

The IMS command issued depends upon operator selection of the "Normal, Immediate or Force" shutdown type on the Operator Shutdown panel and the associated default *shutoption* specified in the SHUTNORM, SHUTIMMED and SHUTFORCE control file entries (refer to "SHUTNORM/SHUTIMMED/SHUTFORCE—Control region" on page 115); operator selection (optional) of a shutdown option on the Operator Shutdown panel will override the default *shutoption* specification.

The *shutoption* parameter coded in the SHUTNORM, SHUTIMMED and SHUTFORCE entries corresponds to the *shutoption* sub-parameter coded in the REPLY or CMD parameters of the SHUTTYPES entry. For instance, consider the entries coded in the example for SHUTNORM/SHUTIMMED/SHUTFORCE—Control region and in the following example. If the operator requests a shutdown type of "Normal," the /CHE DUMPQ command would be issued as defined by the SHUTTYPES control file entry. The SHUTTYPES entry is related to the SHUTNORM entry through their equivalent *shutoption* parameters. If, however, the operator also selects the FREEZE shutdown option, the default *shutoption* parameter of SHUTNORM (in this example, DUMPQ) will be overridden. Then, the command defined by the FREEZE *shutoption* parameter of the SHUTTYPES entry will be issued (in this case, the /CHE FREEZE command).

Refer to the description of the Operator Shutdown panel in the *AOC/MVS IMS Automation Operator's Guide*.

Examples of Usage

Example 1

```
IMSIMSZ  SHUTTYPES,REPLY=(DUMPQ,RETRY5,'/CHE DUMPQ'),
                   REPLY=(BACKUP,RETRY5,'/CHE BACKUP'),
                   REPLY=(FREEZE,RETRY5,'/CHE FREEZE'),
                   REPLY=(PURGE,RETRY5,'/CHE PURGE'),
                   CMD=(NODUMP,,'MVS F &EHKVAR1,STOP'),
                   CMD=(DUMP,,'MVS F &EHKVAR1,STOP')
```

Example 2

```
IMSIMSZ  SHUTNORM,CMD=(PASS1,,'EVIET001 IMSIMSZ,NORM,DUMPQ')

IMSIMSZ  SHUTTYPES,REPLY=(DUMPQ,RETRY5,'/CHE DUMPQ')
                   REPLY=(FREEZE,RETRY5,'/CHE FREEZE')
```

Notice the correlation between the DUMPQ parameter on the SHUTNORM and SHUTTYPES control file entries. The default is used unless an operator issues an override.

Example 3 DB Control Region

```
IMS01C     SHUTTYPES,
             CMD=(FREEZE,,'MVS &EHKVAR7CHE FREEZE'),
             CMD=(PURGE,,'MVS &EHKVAR7CHE PURGE'),
             CMD=(NODUMP,,'MVS F &EHKVAR1,STOP'),
             CMD=(DUMP,,'MVS F &EHKVAR1,DUMP')
```

To override a shutdown, the operator goes to panel EVIKT100.

```
EVIKT100                 IMS Automation: Operator Shutdown
                                                                 Date: 05/22/95
Subsystem Name . . . . . .  IMS10A1_ (? for list)                Time: 17:11:00

Shutdown Type  . . . . . .  1 1  Normal shutdown
                              2  Immediate shutdown
                              3  Force shutdown
                              4  Cancel pending shutdown request

Restart subsystem  . . . .  2 1  Yes
                              2  No
                              3  Control

Broadcast message  . . . .  2 1  Yes
                              2  No


Press F8 for shutdown option, current value:  *

Next start  : none                    Next shutdown   : none

COMMAND ===>
F1=Help     F2=End      F3=Return                              F6=Roll
            F8=Options
```

Figure 15. IMS Automation: Operator Shutdown Panel

To override the default shutdown type specified in the control file, press PF8 to display the menu shown in Figure 16.

```
                                                                 Time: 16:09

Subsystem, Group or Domain . .  IMS01A

       Shutdown Type NORM

             Restart 2    Broadcast 2


Select Shutdown Option:
                  _ 0  No Option      Removes previously defined option
                    1  DUMPQ
                    .  BACKUP         Only valid for XRF
                    3  FREEZE
                    4  PURGE
                    .  DUMP           Only valid for Force Shutdown
                    .  NODUMP         Only valid for Force Shutdown


COMMAND ===>
F1=Help     F2=End      F3=Return                              F6=Roll
```

Figure 16. Options Menu Available from Operator Shutdown Panel

Select the FREEZE option to complete the shutdown override.

SNAPQ—Issue SNAPQ checkpoint command

This entry specifies the IMS command to be issued when a SNAPQ checkpoint is required.

```
subsys  SNAPQ,REPLY=(,RETRYn,'/CHE SNAPQ')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

/CHE SNAPQ
This is the command issued when a SNAPQ checkpoint is required. Code exactly as specified.

Comments and Usage Notes

This IMS command is issued when a SNAPQ checkpoint is required.

Code this entry as described above (specify number of retry attempts).

Examples of Usage

Example

```
IMSIMSZ  SNAPQ,REPLY=(,RETRY5,'/CHE SNAPQ.')
```

STARTn—Define start commands

START*n* defines optional IMS start commands. You can code START*n* for the control region and for message, fast path, and batch message processing regions.

Note: If you code STARTOPTIONS=PARENT for the message, fastpath, or BMP regions, you must code START*n* for IMS Automation to start the regions.

START*n* can be coded as either START1 or START2. Most entries will use START1. START2 is used only for XRF partner regions (the secondary HSBID).

For IMS control region startup:

```
subsys STARTn,CMD=(,,"MVS S jobname,PARM1='AUTO=&EHKVAR1
            ,HSBID=&EHKVAR2'")
```

For message, fast path, and batch message processing regions:

```
subsys STARTn,CMD=(,,'MVS S IMSRDR,MBR=region')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name.

region
The name of the dependent message region to be started.

Comments and Usage Notes

1. The **HSBID=** keyword is applicable only in an XRF environment.
2. The *n* parameter indicates the corresponding HSBID for the start command to be issued for a particular message region.
3. For an XRF environment, in addition to the required START1 entry, a START2 entry must also be coded to indicate message region startup for the secondary HSBID.
4. START*n* is optional for the IMS control region. If you do not code START*n* for the control region, IMS Automation issues the standard MVS start command (MVS S *jobname*,AUTO=Y/N), where *jobname* is the control region jobname.
5. This entry can be used to define a user-tailored IMS control region START command:
   a. Two parameters are passed for an IMS control region; EHKVAR1 for the AUTO=Y/N option, and EHKVAR2 for the HSBID=1/2 option.
   b. If no HSBID parameter is found, the number 1 is used by default.
6. The START command may be any command used to start the IMS control region.

7. If the IMSCNTL entries PARMS1 and PARMS2 are coded, they will be ignored if this command format is used. You should incorporate those PARM*x* entries into the START command.
8. For an IMS control region, make sure that the SUBSYSTEM control file entry includes:

   ```
   SUBSYSTEM STARTCMD=YES
   ```

   Without this required entry, IMS Automation will not start any IMS regions. When STARTCMD=YES, IMS Automation invokes a startup CLIST that checks to see if any START*n* entries are coded for the control region. If so, IMS Automation uses the command specified on START*n*. If not, IMS Automation defaults to the standard MVS start command. See “STARTUP—Issue commands at IMS subsystem startup” on page 126.
9. For message, fast path, and BMP regions, use STARTOPTIONS=PARENT if you want to specify startup commands using START*n*. If you use STARTCMD=YES for these regions and do not code “EVIEI00T,” then IMS Automation does not check for the START*n* control file entries.

Examples of Usage

Example Non-XRF control region

```
IMSIMS2    START1,
           CMD=(,,"MVS S IMS1JOB,AUTO=&EHKVAR1)
```

Example XRF control region

```
IMS10AA    START1,
           CMD=(,,"MVS S IMS1JOB,AUTO=&EHKVAR1,HSBID=&EHKVAR2")

IMS10AA    START2,
           CMD=(,,"MVS S IMS2JOB,AUTO=&EHKVAR1,HSBID=&EHKVAR2")
```

Example Non-XRF message region

```
MSG1IMSZ   START1,CMD=(,,'MVS S IMSRDR,MBR=MSG1IMSZ')
```

Example XRF message region

```
MSG1IMSA   START1,CMD=(,,'MVS S IMSRDR,MBR=MSG1IMSA')
MSG1IMSA   START2,CMD=(,,'MVS S IMSRDR,MBR=MSG2IMSA')
```

START1 starts the normal message region. START2 starts its XRF partner region.

STARTUP—Issue commands at IMS subsystem startup

The STARTUP entry defines the commands to be executed when a startup process is invoked for an IMS subsystem. For a control region, the CLIST EVIEI00I is the only valid value for the command entry. The second and third syntax formats shown below are used to define the START command used for message, fast path, and batch message regions when STARTOPTIONS=PARENT is not coded for these regions.

```
subsys STARTUP,CMD=(,,EVIEI00I)

subsys STARTUP,CMD=(,,'MVS S IMSRDR,MBR=jobname')

subsys  STARTUP,CMD=(,,'IMSCMD &SUBSAPPL /STA REG procname JOBNAME jobname')
(requires IMS V6.1)

subsys STARTUP,CMD=(,,'EVIEI00T')
```

Keyword and Parameter Definitions

subsys
: This is the SA OS/390 subsystem name.

EVIEI00I
: Identifies the CLIST required for startup of an IMS control region. Code exactly as shown.

jobname
: The startup command to be executed.

EVIEI00T
: Identifies the CLIST required for startup of an XRF message region or BMP. This routine performs synchronization and invokes START1 or START2 entries to start a region. See “STARTn—Define start commands” on page 124.

Comments and Usage Notes

⇒ **Required**. STARTUP is a required control file entry for control regions only.

The STARTUP entry must be specified exactly as shown for an IMS/DC control region. For IMS startup, this CLIST will be invoked to perform part of the startup processing.

Examples of Usage

Example

```
IMS10AA   STARTUP,CMD=(,,EVIEI00I)
```

Example 2 IMS message region without STARTOPTIONS=PARENT

```
MSGIMSZA STARTUP,CMD=(,,'MVS S IMSRDR,MBR=MSGIMSZA)
```

| Subsystem startup option available for V6:

```
MSGIMSSB STARTUP,CMD=(,,'IMSCMD IMS10S /STA REG IMS01MPR JOBNAME MSGIMSSB')
```

STOPBMPREGION—Stop batch message regions

This entry defines the command to stop batch message regions. This entry is used only when SHUTOPTIONS=PARENT is *not* specified.

```
IMS Control Region Syntax:
subsys STOPBMPREGION,
       REPLY=(NORMAL,RETRYn,'/STOP REGION &EHKVAR1'),
       REPLY=(ABEND,RETRYn,'/STOP REGION &EHKVAR1 ABDUMP'),
       REPLY=(CANCEL,RETRYn,'/STOP REGION &EHKVAR1 CANCEL')

DB Control Region Syntax:
subsys STOPBMPREGION,
       CMD=(NORMAL,,'command'),
       CMD=(ABEND,,'command'),
       CMD=(CANCEL,,'command'),
```

Keyword and Parameter Definitions

subsys
The SA OS/390 name of the IMS control region.

RETRY*n*
The number of retries.

Comments and Usage Notes

1. NORMAL equates to SHUTNORM for the BMP region.
2. ABEND equates to SHUTIMMED for the BMP region.
3. CANCEL equates to SHUTFORCE for the BMP region.

Examples of Usage

Example 1 IMS Control Region

```
IMSIMSZA     STOPBMPREGION,
             REPLY=(NORMAL,RETRYn,'/STOP REGION &EHKVAR1'),
             REPLY=(ABEND,RETRYn,'/STOP REGION &EHKVAR1 ABDUMP'),
             REPLY=(CANCEL,RETRYn,'/STOP REGION &EHKVAR1 CANCEL')
```

Example 2 DB Control Region

```
IMS01C     STOPBMPREGION,
             CMD=(NORMAL,,'MVS &EHKVAR7STOP REG &EHKVAR1'),
             CMD=(ABEND,,'MVS &EHKVAR7STOP REG &EHKVAR1 ABDUMP'),
             CMD=(CANCEL,,'MVS &EHKVAR7STOP REG &EHKVAR1 CANCEL')
```

STOPFPREGION—Stop fast path regions

This entry defines the commands to stop fast path regions. This entry is used only when SHUTOPTIONS=PARENT is *not* specified.

```
subsys STOPFPREGION,
       REPLY=(NORMAL,RETRYn,'/STOP REGION &EHKVAR1'),
       REPLY=(ABEND,RETRYn,'/PURGE FPREGION &EHKVAR1'),
       REPLY=(CANCEL,RETRYn,'/STOP REGION &EHKVAR1 ABDUMP')
```

Keyword and Parameter Definitions

subsys
The SA OS/390 subsystem name for the control region.

RETRY*n*
The number of retries

Comments and Usage Notes

1. Not applicable for DB control regions.
2. This entry is defined under the IMS control region definition.
3. NORMAL equates to SHUTNORM for the FP region.
4. ABEND equates to SHUTIMMED for the FP region.
5. CANCEL equates to SHUTFORCE for the FP region.

Examples of Usage

Example IMS Fast Path Region without SHUTOPTIONS=PARENT

```
IMSIMSZA    STOPFPREGION,
            REPLY=(NORMAL,RETRY9,'/STOP REGION &EHKVAR1'),
            REPLY=(ABEND,RETRY9,'/PURGE FPREGION &EHKVAR1'),
            REPLY=(CANCEL,RETRY9,'/STOP REGION &EHKVAR1 ABDUMP')
```

STOPREGION—Stop IMS dependent message region

This entry specifies the IMS command required to stop an IMS dependent message region.

```
subsys  STOPREGION,REPLY=(NORMAL,RETRYn,'/STOP REG  &EHKVAR1')
                   REPLY=(ABEND,RETRYn,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2')
                   REPLY=(CANCEL,RETRYn,'/STOP REG &EHKVAR1 CANCEL')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

RETRY*n*
This entry specifies that IMS Automation will retry the operation *n* times.

ABEND
This entry specifies the IMS command required to stop an IMS dependent message region when the normal stop command fails.

CANCEL
This entry specifies the IMS command required to stop an IMS dependent message region when NORMAL and ABEND stop commands fail.

NORMAL
This entry specifies the IMS command required to stop an IMS dependent message region as a normal part of the shutdown process.

Comments and Usage Notes

1. Not applicable for DB control regions.
2. Two variables are available, &EHKVAR1 and &EHKVAR2. &EHKVAR1 is a variable containing the IMS region number of the dependent region that is being stopped. &EHKVAR2 is transaction name that is active.
3. Code this entry exactly as described above.

Examples of Usage

Example

```
IMS10Z  STOPREGION,REPLY=(NORMAL,RETRY9,'/STOP REG &EHKVAR1'),
            REPLY=(ABEND,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2    '),
            REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 CANCEL')
```

STSFILE—Status file

The STSFILE entry describes the partner IMS status file that will be used in this XRF IMS environment. To support an XRF IMS environment, IMS Automation must access the status file of the partner IMS subsystems. This is only required in an XRF environment.

Note: If the environment is such that the master catalogs are not shared, a STSFILE entry is required for both IMS XRF partner domains.

The STSFILE entry should be coded as follows for the status files of the XRF partners:

```
STSFILE partner-domain,DSN=partner-stsfile
```

Keyword and Parameter Definitions

partner-domain
This parameter is the NetView domain ID that is controlling the partner IMS subsystem in an XRF environment.

partner-stsfile
This parameter is the data set name of the status file used by the partner IMS in the partner's domain. This data set will be dynamically allocated during SA OS/390 initialization.

| The status files must be added to the Systems Exclusion Resource Name List from
| GRS (Global Resource Serialization) or any OEM (Other Equipment Manufacturer)
| equivalent.

Comments and Usage Notes

The status files must be defined to NetView. This is done during the installation of SA OS/390. In a non-XRF environment, message EVI709E will be issued for a missing STSFILE entry. This is a normal situation; the message can be ignored.

If the XRF partner is being controlled by another NetView domain ID, that NetView's status file must be defined in a STSFILE entry. You must have a STSFILE entry to match each NetView domain ID that you have specified in an ENVIRON entry (PRTNRDOM parameter) as being a partner domain.

If the partner-domain is coded as this domain then the partner-stsfile should be the same as the one that is allocated to SA OS/390. If it is not the same then message EVI709E will be issued with the explanatory message text:

```
DATASET partner-stsfile cannot be allocated to AOFSTAT
```

Examples of Usage

If the XRF complex were running on domains AOCS6 and AOCS5, the respective STSFILE entries would be as follows:

Example 1

On domain AOCS5 for target domain AOCS6:

```
STSFILE    AOCS6,DSN=AOCS6.NETVIEW.STATUS.FILE
```

Example 2

On domain AOCS6 for target domain AOCS5:

```
STSFILE    AOCS5,DSN=AOCS5.NETVIEW.STATUS.FILE
```

SUBSYSTEM—Define IMS region to SA OS/390

The SUBSYSTEM entry defines an IMS region to SA OS/390.

```
SUBSYSTEM subsys,JOB=procname
                ,PARENT=parent-name
                ,STARTCMD=YES
                ,SHUTDLY=hh:mm:ss
                [,STARTOPTIONS=PARENT]
                [,SHUTOPTIONS=PARENT]
                [,ARMNAME=armname]
                [,DESC=description]
```

Keyword and Parameter Definitions

Unless specified, the definition of the parameter is provided by the primary SA OS/390 SUBSYSTEM entry.

subsys
This is the SA OS/390 subsystem name of the IMS control region. This name may be different from the job name used for the IMS region. Having different names may make operator understanding more or less difficult, depending on the environment; the decision is installation-dependent.

procname
This parameter specifies the IMS subsystem job name.

| **Note:** The length of an IMS subsystem name is limited to 8 characters. For
| further information, refer to “Restrictions” on page 42.

parent-name
This is the subsystem job name which must be active before this subsystem can be started.

STARTCMD=YES
This parameter is required for control regions and must specify YES. For control regions and dependent regions, if this option is coded, a STARTUP entry must also be defined to describe the command to be invoked for IMS startup (refer to “STARTUP—Issue commands at IMS subsystem startup” on page 126). This parameter cannot be coded with STARTOPTION=PARENT.

SHUTDLY=*hh:mm:ss*
For a control region, shutdown delay time should be a long interval (the CLIST is long running) reflecting the relatively long time required for IMS to exit the system. The delay time specified should be longer than the IMS ENVIRON TTIMER parameter.

STARTOPTIONS=PARENT
| This parameter can be coded only on dependent region subsystem entries.
| The parameter must be used for DBRC and DLI regions. The parameter may
| also be used for Message, Fast Path, and Batch Message Processing regions.
| If this parameter is used with Message, Fast Path, or Batch Message
| Processing regions, you must code the STARTn control file entry to define the
| startup command.

| This parameter specifies that this subsystem may be started by its parent subsystem, the IMS control region. IMS Automation will start dependent regions once the control region is active.

SHUTOPTIONS=PARENT
| This parameter can be coded only on dependent region subsystem entries. The parameter must be used for DBRC and DLI regions. The parameter may also be used for Message, Fast Path, and Batch Message Processing regions. This parameter specifies that this subsystem must be shut down by its parent subsystem, the IMS control region. IMS Automation shutdown routines will shut down the region when the control region is shut down, if applicable.

armname
The name of the ARM element of which this subsystem is a member.

Comments and Usage Notes

⇒ **Required**.

Refer to the SA OS/390 customization dialogs to build this control file entry.

Each IMS region defined to and controlled by SA OS/390 must be defined by one SUBSYSTEM entry for that region. Additionally, the DBRC, DLISAS, and dependent MPP, IFP and BMP regions must also be defined to SA OS/390 with the SUBSYSTEM entry if they are to be controlled by IMS Automation.

Examples of Usage

Example 1 Control region for IMSIMSZ

```
SUBSYSTEM  IMSIMSZ,JOB=IMSIMSZ,DESC='Test IMS subsystem',
              PARENT=VTAM,
              SHUTDLY=10:00,
              STARTCMD=YES
```

Example 2 DBRC region for IMSIMSZ

```
SUBSYSTEM  DBRIMSZ,JOB=DBRIMSZ,DESC='DBRC region for IMSIMSZ',
              PARENT=IMSIMSZ,
              SHUTDLY=00:03,
              STARTOPTIONS=PARENT
              SHUTOPTIONS=PARENT
```

Example 3 DLI/SAS region for IMSIMSZ

```
SUBSYSTEM  DLIIMSZ,JOB=DLIIMSZ,DESC='DLI/SAS region for IMSIMSZ',
              PARENT=IMSIMSZ,
              SHUTDLY=00:03,
              STARTOPTIONS=PARENT
              SHUTOPTIONS=PARENT
```

Example 4 Message region for IMSIMSZ

```
SUBSYSTEM  MSGIMSZ,JOB=MSGIMSZ,DESC='Online message region for IMSIMSZ',
              PARENT=IMSIMSZ,
              SHUTDLY=00:03,
              STARTOPTIONS=PARENT,
              SHUTOPTIONS=PARENT
```

Example 5 Message region for IMSIMSZ

```
SUBSYSTEM  MSGIMSZA,JOB=MSGZA,DESC='A MSG REGION',
              PARENT=IMSIMSZ,
              SHUTDLY=00:01,
              STARTCMD=YES
```

Example 6 Fast path region for IMSIMSZ

```
SUBSYSTEM  FPIMSZA,JOB=FPIMSA,DESC='FP REGION FOR Z',
              PARENT=IMSIMSZA,
              SHUTDLY=00:01,
              STARTCMD=YES
```

Example 7 Batch message region for IMSIMSZA

```
SUBSYSTEM  BMPIMSA,JOB=BMPIMSA,DESC='BMP REGION FOR Z',
              PARENT=IMSIMSZA,
              SHUTDLY=00:01,
              STARTCMD=YES
```

Example 8 FDR region for IMSIMSS

```
SUBSYSTEM  IMS10F,
           DESC='FDR for IMS SUBSYSTEM',
           JOBTYPE=MVS,
           JOB=IMS10F,
           PARENT=(JES2),
           STRTDLY=00:05:00,
           RESTARTOPT=ABENDONLY
```

Example 9 CQS region for IMSIMSS

```
SUBSYSTEM  IMS10C,
           DESC='CQS FOR IMS SUBSYSTEM',
           JOBTYPE=MVS,
           JOB=IMS10C,
           PARENT=(JES2),
           STRTDLY=00:05:00,
           SHUTDLY=00:01:00
```

TCO—Issue commands for time-driven procedures

This entry allows commands to be issued to initiate, change, start, or stop time-driven procedures for any IMS operation. This entry is valid for IMS Version 3 or later only.

```
subsys TCO,REPLY=(INIT,,'DFSTCF LOAD DFSTCF  .'),
           REPLY=(SPEC,,'DFSTCF LOAD &EHKVAR1  .'),
           REPLY=(START,,'/START LTERM DFSTCF1  .'),
           REPLY=(STOP,,'/PSTOP LTERM DFSTCFI   .')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

INIT
This entry specifies the IMS command required to start the initial time driven procedure DFSTCF.

SPEC
This entry specifies the IMS command required to change from the current TCO script to a script name entered from the operator interface.

START
This entry specifies the IMS command required to start TCO.

STOP
This entry specifies the IMS command required to stop TCO.

Comments and Usage Notes

1. For more information on TCO, refer to the *IMS Operations Guide*.
2. &EHKVAR1 is the name entered from the IMS Automation operator interface TCO function.

Examples of Usage

Example

```
IMS10  TCO,REPLY=(INIT,,'DFSTCF LOAD DFSTCF  .'),
           REPLY=(SPEC,,'DFSTCF LOAD &EHKVAR1 .'),
           REPLY=(START,,'/START LTERM DFSTCFI  .'),
           REPLY=(STOP,,'/PSTOP LTERM DFSTCFI  .')
```

TCOMEMBERS—Define TCO Members

This definition for the automation control file creates a list of members that appear in a pop-up IMS Automation TCO Member Load panel. The panel is shown in the example below:

```
EVIKMT10       IMS Automation: TCO Member Load                  Page: 1 of 1
                                                                Date: 05/23/94
                                                                Time: 13:01
Subsystem Name . . . . . :  IMS01A

TCO Status . . . . . . . :  Available
        +---------------------------------------------------------------+
Member  |  Select one of the user supplied TCO members                  |
        |  _ STRTLNES        START LINES                                |
Alterna |  _ ASGNTRAN        ASSIGN TRANSACTIONS                        |
        |  _ DISPROG         DISPLAY ACTIVE PROGRAMS                    |
Maximum |                                                               |
        |                                                               |
        |                                                               |
        |                                                               |
        |                                                               |
        |                                                               |
        |                                                               |
        |                                                               |
        | F1=Help    F3=Cancel                                          |
        +---------------------------------------------------------------+
Command ===>
F1=Help     F2=End      F3=Return   F4=IMS Menu                 F6=Roll
```

subsys **TCOMEMBERS,NAME=(***membername***,'***comment***')**

Keyword and Parameter Definitions

membername
This is the 8-character name of the member previously defined in the IMS TCO member library. The library is associated with the DFSTCF DD statement in the IMS start up JCL.

comment
This is a comment, up to 20 characters long.

Comments and Usage Notes

For more information on TCO, refer to the *IMS Operations Guide*.

Examples of Usage

Example

```
IMS01A    TCOMEMBERS,
            NAME=(STRTLNES,'START LINES'),
            NAME=(ASGNTRAN,'ASSIGN TRANSACTIONS'),
            NAME=(DISPROG,'DISPLAY ACTIVE PROGRAMS')
```

THRESHOLDS—IMS error thresholds

The THRESHOLDS entries define the number of times an event must occur for infrequent, frequent, or critical error situations. The operator is notified each time a threshold level is reached. Normally, automated recovery processes are halted when the critical threshold is reached and the operator takes over manual recovery.

If both the RECOVERY and AUTOMATION flags are set to YES, recovery procedures are initiated as required. This process is usually transparent to the operator. However, operators should be notified when repeated recovery procedures exceed expected limits, in case operator intervention is needed.

```
THRESHOLDS name
           ,CRIT=(number,interval)
           ,FREQ=(number,interval)
           ,INFR=(number,interval)
```

Keyword and Parameter Definitions

name
: The name by which this IMS subsystem is known to SA OS/390. The name is defined with the SUBSYSTEM control file entry. You can append the name of a minor resource to the subsystem name to code a THRESHOLDS entry for just that minor resource. Valid minor resource names are:

MSC	All Multiple Systems Couplers (MSC) links on the IMS
MSC.*link_id*	A single MSC link
OLDS	All online data sets (OLDS)
PROG	All program-driven BMP failures
PROG.*prog_id*	A single program-driven BMP failure
TRAN	All transaction failures
TRAN.*trans_id*	A single transaction failure

CRIT=*(number,interval)*
: Specifies the event thresholds for critical situations. Normally, recovery is terminated if a situation reaches critical condition. This is done to prevent problems with potentially recursive situations.

  number is the number of occurrences (from 1 to 10) that defines the condition.

  interval is the time period in hours and minutes (*hh:mm*) that defines the condition. The range is from 00:01 to 99:59.

FREQ=*(number,interval)*
: Specifies event thresholds for frequent occurrences.

INFR=*(number,interval)*
: Specifies event thresholds for infrequent occurrences.

Comments and Usage Notes

1. Refer to the SA OS/390 customization dialogs to build this control file entry.
2. If you set a threshold level to 1, the threshold will be exceeded on the first occurrence of an error.
3. These minor resource threshold entries must be entered directly into the %INCLUDE member. The SA OS/390 customization dialog cannot be used for these threshold entries.
4. The threshold entries should uniquely describe consistent intervals for processing, with the critical thresholds having either the smallest interval or the highest frequency of occurrence, followed by frequent, then by infrequent conditions.

Examples of Usage

Example subsys

```
THRESHOLDS IMS10AA,CRIT=(2,08:00),FREQ=(1,04:00),INFR=(1,24:00)
```

Example subsys.TRAN

```
THRESHOLDS IMS10AA.TRAN,CRIT=(2,00:45),FREQ=(2,04:00),INFR=(2,24:00)
```

Example subsys.TRAN.transid

```
THRESHOLDS IMS10AA.TRAN.SAMPLE1,
           CRIT=(2,00:45),FREQ=(2,04:00),INFR=(2,24:00)
```

SAMPLE1 is the transaction ID.

Example subsys.PROG

```
THRESHOLDS IMS10AA.PROG,CRIT=(2,00:45),FREQ=(2,04:00),INFR=(2,24:00)
```

Example subsys.PROG.progid

```
THRESHOLDS  IMS10AA.PROG.SAMPLE2,
            CRIT=(2,00:45),FREQ=(2,04:00),INFR=(2,24:00)
```

SAMPLE2 is the program ID.

Example subsys.OLDS

```
THRESHOLDS IMS10Z.OLDS,
           CRIT=(05,00:30),FREQ=(03,00:30),INFR=(01,00:30)
```

TPABEND—Dependent region abend

This entry is required for a dependent region abend with a U0002 return code.

```
subsys   TPABEND,CODE=(U0002,,,ABENDING)
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

Comments and Usage Notes

If a dependent region abend with a U0002 return code occurs, all active MPP and BMP dependent regions are forced to terminate, and are prevented from restarting before the control region goes to abend status.

Examples of Usage

Example

```
IMSIMSZ  TPABEND,CODE=(U0002,,,ABENDING)
```

TRIGGER—Startup and shutdown triggers

Use this entry to define startup and shutdown trigger conditions.

```
TRIGGER subsys
        [,STARTUP={(SERVICE,event,...event)|(event,...event)}]
        .
        .
        .
        [,STARTUP={(SERVICE,event,...event)|(event,...event)}]
        .
        .
        .
        [,SHUTDOWN={(SERVICE,event,...event)|(event,...event)}]
        .
        .
        .
        [,SHUTDOWN={(SERVICE,event,...event)|(event,...event)}]
```

Keyword and Parameter Definitions

subsys
The symbolic name by which this IMS subsystem is known to SA OS/390, as defined with the SUBSYSTEM control file entry.

STARTUP=
Use STARTUP= to define startup triggers and conditions, where:

SERVICE
Indicates that a startup cannot occur unless it is within a service period window. (Refer to “SERVICE—Service period definitions” on page 112.)

event
Is the symbolic name of the external condition that represents this trigger.

SHUTDOWN=
Use SHUTDOWN= to define shutdown triggers and conditions, where:

SERVICE
Indicates that a shutdown cannot occur unless the service period window is closed.

event
Is the symbolic name of the external condition that represents this trigger.

Comments and Usage Notes

1. Refer to “IMSPOST—Post an external event” on page 172 to see how triggers are set and unset.
2. Multiple startup or shutdown triggers can be specified. To indicate that all triggers must be set, include them in the same STARTUP= or SHUTDOWN= entry. To indicate either/or conditions, use multiple STARTUP= or SHUTDOWN= entries. Refer to examples 3 and 4.
3. Up to 28 triggers can be specified for each subsystem.

Examples of Usage

Example 1

```
TRIGGER IMS10AA,STARTUP=(SERVICE,BATCH1)
```

This example indicates that the startup process is initiated as soon as both conditions are met: the trigger named BATCH1 is set and a service period window is open.

Example 2

```
TRIGGER IMS10AA,STARTUP=BATCH1
```

In this example, the startup process is initiated when the trigger named BATCH1 is set, regardless of whether or not the service period window is open.

Example 3

```
TRIGGER IMS10AA,STARTUP=(SERVICE,BATCH1),
              STARTUP=(SERVICE,BATCH2)
```

This example defines two situations in which a startup is initiated. In the first situation, BATCH1 must be set and a service period window must be open. In the second situation, BATCH2 must be set and a service period window must be open. A startup is initiated when the conditions for either situation are satisfied.

Also, a startup can not be initiated unless either BATCH1 or BATCH2 is set when a service period window does open.

Example 4

```
TRIGGER IMS10AA,SHUTDOWN=(BATCH3,BATCH4)
```

In this example, a shutdown is initiated as soon as both triggers (BATCH3 and BATCH4) are set, regardless of the service period window.

UNLKAVM—Reply to message AVM005A

This command is used in reply to the outstanding reply of message AVM005A to inform the alternate that input/output prevention is complete.

```
subsys  UNLKAVM,CMD=(,,'MVS R &EHKVAR1,UNLOCK')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

Comments and Usage Notes

Required only for an XRF IMS subsystem.

This IMS command is issued as part of the XRF takeover process. This entry will normally be specified as described above.

The variable &EHKVAR1 contains the outstanding reply that is associated with the AVM manager AVM005E.

Examples of Usage

Example

```
IMSIACOA   UNLKAVM,CMD=(,,'MVS R &EHKVAR1,UNLOCK')
```

UNLOCK—Input/output prevention completed

This command informs the alternate XRF subsystem that input/output prevention is complete.

```
subsys  UNLOCK,REPLY=(,RETRYn,'/UNLOCK SYSTEM.')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

RETRY*n*
n specifies the number of retry attempts.

Comments and Usage Notes

Required only for an XRF IMS subsystem.

This IMS command is issued as part of the XRF takeover process. This entry will normally be specified as described above, where *n* is the number of retry attempts.

Examples of Usage

Example

```
IMSIAOCA   UNLOCK,REPLY=(,RETRY5,'/UNLOCK SYSTEM.')
```

UP—Initiate FDR Startup

This entry defines the Automation procedure that will be run to initate the startup of the FDR region when the IMS Control region changes to 'UP' status.

```
subsys UP,CMD=(,,EVISTFDR)
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

Comments and Usage Notes

This is only applicable to FDR enabled IMS regions.

When the IMS Control region status changes to 'UP', Automation will run REXX procedure EVISTFDR to initiate the startup of the FDR region on the NetView domain as specified in the Automation policy.

Examples of Usage

Example

```
IMS10S  UP,CMD=(,,EVISTFDR)
```

USERSTART—User-defined restart commands

This entry defines customer-specified IMS restart commands to be integrated into the operator interface.

```
subsys USERSTART,restart_name='restart_command'...
```

Keyword and Parameter Definitions

subsys
This is the name of the IMS subsystem to be started.

restart_name
Name for the IMS restart command defined by the system programmer.

restart_command
Actual IMS restart command to be issued in response to the DFS810A message (for IMS control regions) or the DFS989I message (for DB control regions).

Comments and Usage Notes

The IMS Automation code will automatically supply the required slash (/) and final period for the IMS restart command.

Examples of Usage

Example

```
IMS10AA  USERSTART,ERE1='ERE COLDCOMM',
                   NRE1='NRE CHKPT 0'
```

VTAMTERMS—Release terminal sessions

This entry specifies the command IMS Automation will issue to release terminal sessions with IMS, accelerating IMS shutdown and XRF IMS takeover.

```
subsys VTAMTERMS,
       CMD=(SHUTDOWN,,'V NET,TERM,LU1=&EHKVAR1,TYPE=UNCOND'),
       CMD=(TAKEOVER,,'V NET,TERM,LU=&EHKVAR1,TYPE=FORCE')
```

Keyword and Parameter Definitions

subsys
This is the IMS subsystem to be shut down or taken over.

SHUTDOWN
This entry specifies the command IMS Automation will issue to release terminal sessions with IMS and accelerate IMS shutdown.

TAKEOVER
This entry specifies the command IMS Automation will issue to release terminal sessions with IMS to accelerate XRF IMS takeover.

Comments and Usage Notes

Required for an XRF IMS subsystem.

This MVS command is issued to release terminal sessions with IMS to accelerate IMS shutdown and XRF IMS takeover.

The variable &EHKVAR1 contains the actual VTAM LU name.

Examples of Usage

Example for non-XRF

```
IMS10AA VTAMTERMS,CMD=(SHUTDOWN,,'V NET,TERM,LU1=&EHKVAR1,TYPE=UNCOND')
```

Example for XRF

```
IMS10AA VTAMTERMS,CMD=(SHUTDOWN,,'V NET,TERM,LU1=&EHKVAR1,TYPE=UNCOND'),
                  CMD=(TAKEOVER,,'V NET,TERM,LU1=&EHKVAR1,TYPE=FORCE')
```

%INCLUDE—Include additional members in control file

Use the %INCLUDE entry to include additional members when the control file is loaded. This allows the control file to be maintained by the appropriate system group responsible for their particular business function. It also allows you to split those members that are built using the SA OS/390 customization dialog from those members that are edited.

```
%INCLUDE member-name
```

Keyword and Parameter Definitions

member-name
The name of the member to be included into the control file when the control file is loaded. The member must reside on a data set referenced by DSIPARM.

Example

This examples shows includes for three subsystems. The member names are IMS10AA, IMS10AB, and IMS10Z.

```
%INCLUDE IMS10AA
%INCLUDE IMS10AB
%INCLUDE IMS10Z
```

$PI—Stop BMP tasks on active subsystem

This entry specifies the commands required to stop BMP tasks on the old *active* IMS subsystem's CPC during takeover, so that the BMP workload can be transferred to the *alternate* IMS subsystem's (new *active*) CPC.

```
subsys  $PI,CMD=(,,'MVS $PIm-n')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

m-n
The range of initiator numbers to be stopped.

Comments and Usage Notes

⇒ **Not required**. Applicable to an XRF environment only.

These MVS/JES commands are issued to stop BMP initiators on the old *active* on a takeover so that BMP workload can be transferred to the system where the new *active* is running.

Examples of Usage

Example

```
IMSIAOCA   $PI,CMD=(,,'MVS $PI1-12')
```

This entry stops initiators 1 through 12.

$SI—Issue start initiator commands after XRF takeover

These start initiator commands are issued after an XRF takeover is complete, immediately following the $TI commands.

```
subsys  $SI,CMD=(,,'MVS $SIm-n')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

m-n
The range of initiator numbers to be started.

Comments and Usage Notes

⇒ **Not required**. Applicable to an XRF environment only.

These MVS/JES commands are issued after an XRF takeover is complete. They are issued on the system of the new *active*. While you can code any MVS/JES commands for this entry, it is specifically intended as a means to adjust the initiator structure after an XRF takeover.

Examples of Usage

Example

```
IMSIAOCB   $SI,CMD=(,,'MVS $SI1-12')
```

This entry starts initiators 1 through 12.

$TI—Assign job classes after XRF takeover

These commands are issued after an XRF takeover is complete, immediately preceding the $SI commands, in order to assign the job classes which an initiator is to process.

```
subsys  $TI,CMD=(,,'MVS $TIm-n,jobclass1{,jobclass2...}')
```

Keyword and Parameter Definitions

subsys
This is the SA OS/390 subsystem name of the IMS control region.

m-n
The range of initiator numbers to which specified job classes are to be assigned.

jobclassn
The job classes to assign to these initiators.

Comments and Usage Notes

⇒ **Not required**. Applicable to an XRF environment only.

This entry is similar to the $SI entry. The commands specified here are issued immediately preceding the $SI commands.

Examples of Usage

Example

```
IMSIAOCA   $TI,CMD=(,,'MVS $TI1-12,A')
```

Chapter 4. Common Routines

This part describes IMS Automation common routines which request information or perform tasks associated with IMS Automation. You can use these common routines in automation procedures you create. Examples, sample routines, and data area information are given to show how this might be done.

IMS Automation provides new routines to retrieve and update IMS Automation-unique information. These routines can also be used in user-written extensions of IMS Automation. The following routines are arranged alphabetically for easy reference.

EVIEX002—Retrieve XRF partner data

The purpose of this command is to retrieve data from the IMS status file extension record.

```
EVIEX002  subsystem,keyword1,...keywordN
```

Keyword and Parameter Definitions

subsystem
The name of the subsystem for which the information will be retrieved.

keywordN
The name of the keyword of the data to return. The possible values are:

ALTSYS, AVM, BUILDQCKPT, BUILDQHSB, DBRCRESRV, DCSTATUS, DEPREGID, EDNDT, LASTABENDCODE, GENAPPLID, LHSBID, LOGCKPT, RSTCHKPT, RSTHSBID, RUNSTARTYPE, SERVSTARTDT, SERVENDDT, STARTDT, STARTHSB, STARTOPT, STARTTYPE, TOCODE, VER, XRFMODE, XRFSTATUS.

For more information about the keywords, refer to common routine EVIEX003.

Comments and Usage Notes

When the EVIEX002 command is issued, the values will be returned in the following task global variables:

1. EVISTSV*n*

   These variables will contain the value of the keywords in the order in which they were requested. These variables are used for any number of keywords requested. If the keyword requested is invalid, the value returned will be blanks.

2. EVISTSN

   This variable contains the value of the number of keywords requested.

Examples of Usage

Example 1

Coding this:

```
EVIEX002 IMSIAOCA,XRFSTATUS
```

assigns these values:

EVISTSV1 Value of the XRFSTATUS data field

EVISTSN 1

Example 2

Coding this:

```
EVIEX002 IMSIAOCA,XRFSTATUS,ALTSYS,XRFMODE
```

assigns these values:

EVISTSV1 Value of the XRFSTATUS data field

EVISTSV2 Value of the ALTSYS data field

EVISTSV3 Value of the XRFMODE data field

EVISTSN 3

Example 3

The following example shows how to call EVIEX002 from within a REXX CLIST. The results (as they would appear at a NCCF operator session) are displayed after the sample CLIST.

```
/*******************************************************************/
 TRACE OFF
 SAY 'THIS IS AN EXAMPLE OF A CALL TO COMMON ROUTINE EVIEX002'
 'EVIEX002 IMS05Z,STARTTYPE,RSTCKPT'
 'GLOBALV GETT EVISTSV1 EVISTSV2 EVISTSV3 EVISTSN'
 SAY 'EVISTSV1 VARIABLE IS: ' EVISTSV1
 SAY 'EVISTSV2 VARIABLE IS: ' EVISTSV2
 SAY 'EVISTSN VARIABLE IS:  ' EVISTSN
       RETURN
```

Note:

```
Sample output
  THIS IS AN EXAMPLE OF A CALL TO COMMON ROUTINE EVIEX002
  EVISTSV1 VARIABLE IS:  AUTO
  EVISTSV2 VARIABLE IS:  93301/081251
  EVISTSN VARIABLE IS:  2
```

EVIEX003—Update status file data

The purpose of this command is to update data on the IMS status file extension record.

```
EVIEX003 subsystem,keyword1=value1,...keywordN=valueN
```

Keyword and Parameter Definitions

subsystem
The name of the subsystem for which the information will be updated.

keywordN
The name of the field to be updated.

valueN
The value of the field to be updated.

The following keywords can be specified:

ALTSYS=
Identifies the status of the alternate system in an XRF environment.

OFF The alternate system is inactive
ON The alternate system is active

AVM=
Indicates if the Availability Manager is available. The Availability Manager is the controller of an XRF system that determines when a partner or backup system will takeover.

OFF Availability Manager not available
ON Availability Manager available

BUILDQCKPT=
Specifies the last checkpoint ID that is valid for loading DC blocks.

The value is extracted from message DSF3804I, which indicates the completion of an IMS checkpoint.

The format is yyddd/hhmmss.

BUILDQHSB=
When running in an XRF environment, this specifies which IMS system (HSBID) wrote the CHKPT. The value is extracted from message DSF3804I, which indicates the completion of an IMS checkpoint and the IMS system (HSBID) which wrote the CHKPT.

DBRCRESRV=
Specifies the DBRC (IMS Database Recovery Control) reserve indicator. This will tell the old active that a reserve has been detected. The old active will then cancel its DBRC to free the reserve, allowing the new active to proceed with the takeover. The possible values are RESERVE or blank.

DCSTATUS=
Specifies the value of the Data Communication Status field.

The possible values are UP, STOPPED, or DOWN.

DEPREGID=
Exists only for dependent regions; specifies the ID of the dependent processing region: Fast Path, Message, or Batch.

ENDDT=
Specifies the date and time that the IMS subsystem terminated. Normally, this is when IMS shutdown message DFS994I has been processed.

The format is dd/mm/yy,hh:mm:ss.

The value is set to `'--/--/--,--:--:--'`, when the termination of the IMS subsystem is initiated (due to the fact that a shutsys command has been issued, or when an IMS subsystem is about to abend and message DFS629I is being processed).

LASTABENDCODE=
Specifies the abend code for the last abnormal termination of the particular IMS subsystem.

GENAPPLID=
Specifies the generic VTAM applid used during the current or last execution of the particular IMS subsystem.

LHSBID=
When running in an XRF environment, this identifies the last-active IMS system.

The possible values are 1 or 2.

LOGCKPT=
When running in an XRF environment, this indicates the checkpoint at which IMS starts reading the input log.

This is triggered by message DFS3804I in the format of `'LOG READ CHKPT: yyddd/hhmmss'`. This form of the message is issued only after a takeover occurred, and when the system checkpoint needed for IMS recovery was written before the LAST DC CHKPT and before the takeover.

The format of the ID is yyddd/hhmmss.

RSTCKPT=
When running in an XRF environment, this indicates the restart checkpoint ID.

The format of the ID is yyddd/hhmmss.

RSTHSBID=
When running in an XRF environment, this indicates which IMS system (HSBID) wrote the checkpoint.

RUNSTARTTYPE=
Specifies the last requested start type. The field is populated after receiving message DFS629I, DFS627I, or IEF450I, indicating that an IMS has terminated.

The possible values are:

AUTO
The IMS system has been started with 'AUTO=Y'.

BUILDQ
The IMS system has been started using the BUILDQ entry from the control file.

A BUILDQ start implies that all messages on the queues are to be saved across restarts.

COLD
The IMS system has been COLD started.

MANUAL
The IMS system has been started using IMS subsystem RESTART command, entered by the user.

WARMSDBL
The IMS system has been started by loading the Main Storage Date Base (MSDB) during a WARM start.

SERVSTARTDT=
Specifies the time and date when IMS startup message DFS994I was processed.

When running in an XRF environment, this specifies the date and time that the alternate IMS system started. At the same time, the XRFMODE is set to 'BACKUP'.

The format is dd/mm/yy,hh:mm:ss.

The value is set to '--/--/--,--:--:--' during IMS startup when message DFS3410I is received.

SERVENDDT=
Specifies the date and time when the termination of the IMS subsystem has been initiated (due to the fact that a shutsys command has been issued, or when an IMS subsystem is about to abend and message DFS629I is being processed).

The format is dd/mm/yy,hh:mm:ss.

STARTDT=
Specifies the time and date when message DFS3410I has been received during IMS startup. At the same time, SERVSTARTDT is set to a value of '--/--/--,--:--:--'.

The format is dd/mm/yy,hh:mm:ss.

STARTHSB=
When running in an XRF environment, this specifies the HSBID used to start the IMS system.

The possible values are 1 or 2.

STARTOPT=
When running in an XRF environment, this indicates whether the IMS system is meant to be started as the active or the alternate IMS system.

ACTIVE
Start as the active IMS system.

XRF
This is treated like a value of ACTIVE.

BACKUP
Start as the alternate IMS system.

NOXRF
IMS system is not part of an XRF complex.

STARTTYPE=
Specifies the start type to be used to start the IMS system. The possible values are:

AUTO
The IMS system will be started with `'AUTO=Y'`.

BUILDQ
The IMS system will be started using the BUILDQ entry from the control file.

A BUILDQ start implies that all messages on the queues are to be saved across restarts.

COLD
The IMS system will be COLD started using the `'COLD'` control file entry.

MANUAL
The IMS system will be started using RESTART commands, entered by the user.

WARMSDBL
The IMS system will be started by loading the Main Storage Date Base (MSDB) during a WARM start.

TOCODE=
When running in an XRF environment, this specifies the reason code for the takeover. The reason code is extracted from messages DFS3890I or DFS3869I when the takeover processing begins.

Possible values are:

Code (Hex)	Meaning
80	VTAM termination
40	RDS surveillance
20	System LNK surveillance
10	LOG surveillance
08	/SWITCH entered
04	IRLM failure
02	ESTAE exit entered

VER=
Specifies the IMS product executing in the particular IMS subsystem, for instance:

Value	Meaning
2.2	IMS/ESA Version 2 Release 2
3.1	IMS/ESA Version 3 Release 1
4.1	IMS/ESA Version 4 Release 1
5.1	IMS/ESA Version 5 Release 1
6.1	IMS/ESA Version 6 Release 1

XRFMODE=
When running in an XRF environment, this identifies the IMS system as the ACTIVE or BACKUP partner in the XRF pair.

XRFSTATUS=
When running in an XRF environment, this identifies the status of the IMS subsystem. Possible values are:

ACTIVE
Indicates that this is the ACTIVE partner in the XRF pair. The status change is triggered by message DFS0488I, indicating that the UNLOCK command has completed.

ACTIVEDOWN
Indicates that the IMS subsystem which ended was the ACTIVE partner in the XRF pair.

BACKUP
Indicates that this is the ALTERNATE partner in the XRF pair. The status change is triggered by message DFS3838I or DFS3839I, indicating that the initial DB/DC PRE-OPEN has completed.

BACKUPDOWN
Indicates that the IMS subsystem which ended was the BACKUP partner in the XRF pair.

WAITBACKUP
Indicates that the alternate partner is being shutdown.

UP
Valid only for alternate IMS message regions and only when the SYNCH parameter is set to a value of `'S'`.

The SYNCH parameter can be specified in the ENVIRON entry of the Automation Control File and controls the type and degree of region synchronization between the active and the alternate IMS subsystem.

Please refer to the ENVIRON SETUP description in Chapter 3, “Programming Information” on page 41 for more information.

When the automation status of the active partner’s message region is UP, the XRFSTATUS of the alternate message region will be set to UP.

STOPPED
Valid only for alternate IMS message regions and only when the SYNCH parameter is set to a value of `'S'`.

The SYNCH parameter can be specified in the ENVIRON entry of the Automation Control File and controls the type and degree of region synchronization between the active and the alternate IMS subsystem.

Please refer to the ENVIRON SETUP description in Chapter 3 for more information.

When the automation status of the active partner’s message region is not UP, the XRFSTATUS of the alternate message region will be set to STOPPED.

DOWN
Valid only for alternate IMS message regions and only when the SYNCH parameter is set to a value of `'S'`.

The SYNCH parameter can be specified in the ENVIRON entry of the Automation Control File and controls the type and degree of region synchronization between the active and the alternate IMS subsystem.

Please refer to the ENVIRON SETUP description in Chapter 3 for more information.

When a message region of the active region terminates, and message DFS552I is received, the XRFSTATUS of the alternate's message region will be set to DOWN.

NOXRFC
During startup of an IMS system, message DFS3802W, DFS3898W, or DFS3899W have been received. The messages indicate that the system being started will not be XRF capable. This may be as expected in the case of an IMS system genned with XRF but not using it at present.

When message DFS3802 is received, the IMS system is terminated if it is an alternate. Otherwise, startup will continue.

IOPREV
On the active system (which is about to become the old active system), a takeover condition has been reached. The takeover should also start soon on the alternate subsystem.

IOFREE
The IOFREE status means that I/O Prevention has completed on the old active subsystem. This is the status needed on the old active system before the I/O Toleration process can be completed on the new active (by issuing the /UNLOCK command).

TAKEOA
Indicates that IMS takeover is in progress. All class 2 sessions will be inactivated on the old active system. This is triggered by message DFS3891I, indicating that the IMS system log from the active system has been processed by the alternate system.

TAKEOB
The alternate system has initiated a takeover and is about to become the new active.

IOT
This is the status of the new active subsystem after a takeover, during the I/O Toleration process, and prior to becoming a true active system.

Comments and Usage Notes

When the EVIEX003 command is issued, the values for the named subsystem will be changed in the status file.

No validation is performed on the values you specify. If you specify incorrect values, unpredictable results can occur.

Examples of Usage

Example 1

If the following statement were coded:

```
EVIEX003 IMSIAOCA,XRFMODE=ACTIVE
```

the IMS status extension record for the subsystem IMSIAOCA would be updated to reflect the value of ACTIVE for the data field XRFMODE.

Example 2

The following example shows how to call EVIEX003 from within a REXX CLIST. The results can be displayed by calling the common routine EVIEX002. The command results (as they appear in an operator NCCF session) are shown after the program example.

```
/*******************************************************************/
TRACE OFF
SAY 'THIS IS AN EXAMPLE OF A CALL TO COMMON ROUTINE
EVIEX003 & EVIEX002'

'EVIEX003 IMS05Z,XRFMODE=ACTIVE'       /* CHANGES VALUE OF
                                          XRFMODE to 'ACTIVE' */

'EVIEX002 IMS05Z,XRFMODE'              /* RETRIEVE NEW VALUE
                                          OF XRFMODE          */

'GLOBALV GETT EVISTSV1 EVISTSN'

SAY 'EVISTSV1 VARIABLE IS: ' EVISTSV1  /* DISPLAY VALUE       */
SAY 'EVISTSN VARIABLE IS:  ' EVISTSN

'EVIEX003 IMS05Z,XRFMODE=BACKUP'       /* RESET VALUE OF
                                          XRFMODE             */

'EVIEX002 IMS05Z,XRFMODE'              /* RESET VALUE OF
                                          XRFMODE             */
'GLOBALV GETT EVISTSV1 EVISTSN'
SAY 'EVISTSV1 VARIABLE IS: ' EVISTSV1  /* DISPLAY VALU E      */
RETURN
```

Note:

```
Sample output
   C AOF05    EVISTSV1 VARIABLE IS:  ACTIVE
   C AOF05    EVISTSN VARIABLE  IS:  1
   C AOF05    EVISTSV1 VARIABLE IS:  BACKUP
```

EVIEX004—Retrieve XRF partner data

The purpose of this command is to retrieve data from the IMS status file extension record of the partner subsystem in an XRF environment.

```
EVIEX004  partner_subsystem,partnerdom,keyword1...,keywordN
```

Keyword and Parameter Definitions

partner_subsystem
The name of the partner subsystem for which the information will be retrieved.

partnerdom
The domain ID of the partner's NetView that is controlling the partner subsystem.

keywordN
| The name of the field to be returned. The possible values are:

| ALTSYS, AVM, BUILDQCKPT, BUILDQHSB, DBRCRESRV, DCSTATUS,
| DEPREGID, EDNDT, LASTABENDCODE, GENAPPLID, LHSBID, LOGCKPT,
| RSTCHKPT, RSTHSBID, (RUNAPPLID), RUNSTARTYPE, SERVSTARTDT,
| SERVENDDT, STARTDT, STARTHSB, STARTOPT, STARTTYPE, TOCODE,
| VER, XRFMODE, XRFSTATUS.

| For more information about the keywords, refer to common routine EVIEX003.

Comments and Usage Notes

When the EVIEX004 command is issued, the values for the partner subsystem will be returned in the following task global variables:

1. EVISTSVn

   These variables will contain the value of the keywords in the order in which they were requested. These variables are used for any number of keywords requested. If the keyword requested is invalid, the value returned will be blanks.

2. EVISTSN

   This variable contains the value of the number of keywords requested.

Examples of Usage

Example 1

If you code this:

|
```
EVIEX004 IMSIAOCA,CNM02,XRFSTATUS
```

the request will retrieve the data from the status file in domain CNM02 and the following values will be assigned:

| **EVISTSV1** Value of the XRFSTATUS data field for the partner subsystem in
| domain CNM02

EVISTSN 1

Example 2

Coding this:

```
EVIEX004 IMSIAOCA,CNM02,XRFSTATUS,ALTSYS,XRFMODE
```

Assigns these values:

EVISTSV1 Value of the XRFSTATUS data field for the partner subsystem in domain CNM02

EVISTSV2 Value of the ALTSYS data field for the partner subsystem in domain CNM02

EVISTSV3 Value of the XRFMODE data field for the partner subsystem in domain CNM02

EVISTSN 3

Example 3

The following example shows how to call EVIEX004 from within a REXX CLIST. The results (as they appear on an operator NCCF session) follow the program sample.

This common routine can only be used for an XRF environment running in a dual CEC configuration.

```
/*********************************************************************/
 TRACE OFF
/* SAMPLE REXX EXEC                                                   */
 SAY 'THIS IS AN EXAMPLE OF A CALL TO COMMON ROUTINE EVIEX004'

 'EVIEX004 IMS01E,AOF01,AVM,DCSTATUS'

 'GLOBALV GETT EVISTSV1 EVISTSV2 EVISTSN'

 SAY 'EVISTSV1 VARIABLE IS: ' EVISTSV1
 SAY 'EVISTSV2 VARIABLE IS: ' EVISTSV2

 SAY 'EVISTSN VARIABLE IS: ' EVISTSN
 RETURN
```

Note:

```
Sample output
C AOF10    THIS IS AN EXAMPLE OF A CALL TO COMMON ROUTINE EVIEX004
C AOF10    EVISTSV1 VARIABLE IS:  OFF
C AOF10    EVISTSV2 VARIABLE IS:  DOWN
C AOF10    EVISTSN VARIABLE IS:  2
```

IMSBMSG—Build message processor

This processor allows programs written in REXX, NetView command lists, and high level languages to build a single or multi-line message and have it delivered to a designated task on any connected NetView domain.

```
IMSBMSG START,domainid,taskid
IMSBMSG DATA,linetype,data
IMSBMSG END
IMSBMSG CANCEL
```

The parameters are positional.

START
This parameter is used for the first call to IMSBMSG. Parameters on the START indicate the target of the message or messages, where *domainid* is the identifier of the domain to which the message is to be routed and *taskid* is the identifier of the task on that domain. The *taskid* can be a real operator name or the name of an autotask on the receiving system.

DATA
This identifies a user message being passed to the IMSBMSG processor where *line type* and *data* is the message that is to be passed to the target task to which the message is to be routed. The line type of the element of a multi-line message, valid values, and their meanings are as follows.

C Control line
L Label line
D Data line
E End line
F End-with-data line
S Single line.

If a multi-line message is to be issued, a valid combination of line types must be passed.

END
Indicates to the IMSBMSG processor that processing is complete. It causes the processor to route previously issued data lines to the target indicted on the IMSBMSG START command. All storage obtained during processing is released. An IMSBMSG START must be issued before any other DATA items.

CANCEL
This indicates to the IMSBMSG processor that processing is not to continue. All storage is released without any message being issued. Use CANCEL to abort the message.

Comments and Usage Notes

1. All calls must be made from the same invocation of the same program.
2. A correct sequence of line types should be passed. However, if the processor detects that a multi-line message is being passed and no E or F type has been received by the time the IMSBMSG END is received, then message **EVI698 END generated** is added to the group.

3. Single-line and multi-line messages can be mixed in the same START-END bracket.

4. The return codes are:

0	OK.
4	Invalid type. Type should be START, DATA, END or CANCEL.
8	Invalid line type. Type should be C, L, D, E, F, or S.
12	Execution failed. A message is issued that describes the failure.
16	DATA, END or CANCEL issued without START.
20	START issued while previous start in effect.
24	Wrong number of parameters.
28	Invalid domainid. Ensure the domainid specified is correct.
32	The domain identified by domainid is not active.
36	The taskid specified is invalid or not active.
40	The value of the CGLOBAL EVI_BASE_PRODUCT is NULL or invalid. Ensure that the IMS feature Exit EVIEE001 has run and completed successfully.

Examples of Usage

Example

The following example shows how to call IBMBMSG from within a REXX CLIST. The results (as shown on an operator NCCF session) follow the code sample.

```
/**********************************************************************/
 TRACE OFF
 TRACE OFF
SAY 'THIS IS AN EXAMPLE OF A CALL TO COMMON ROUTINE IMSBMSG'

'IMSBMSG START,AOF05,ROMAN'
'IMSBMSG DATA,S,This is a test................. '
'IMSBMSG END'

SAY 'RC FROM IMSBMSG IS : ' RC

 RETURN
```

Note:

```
Sample output
* AOF10    CLISTX
C AOF10    THIS IS AN EXAMPLE OF A CALL TO COMMON ROUTINE IMSBMSG
U AOF10    THIS IS A TEST..................
C AOF10    RC FROM IMSBMSG IS :  0
```

Note: Data is echoed back to caller.

IMSCMD—Issue IMS commands

This common routine is used to perform the requested function on the domain where the named IMS resides, whether local or remote. The calling program does not have to be aware of where the IMS resides. It is particularly useful with single-point-of-control as IMSCMD first determines the domain in which the subsystem resides before building and issuing the request. It then either calls the requested function if the subsystem is on the local domain, or it forwards the command to the remote domain, thus allowing cross-domain communications.

```
IMSCMD subsystem IMS_command
```

Keyword and Parameter Definitions

subsystem
The IMS subsystem that can be controlled by the domain. It can be either a local or a remote IMS subsystem.

IMS_command
The actual IMS command to be issued.

Comments and Usage Notes

1. Security is checked as described in "IMSSEC—Invoke security checking" on page 179.
2. Results from a command are returned in a multi-line message.
3. The IMS command must be the full IMS command or a subset of the characters. For example, /CHECKPOINT and /CHE are valid. Command synonyms are not supported, such as /CHKPT for /CHECKPOINT. Use of command synonyms will result in message EVI501 with return code 110.
4. The IMS commands are restricted to those commands allowed for an IMS automated operator program. This list of commands is documented in the *IMS Customization Guide* under the heading "Automated Operator Commands." The list is also documented in the *IMS Operators Guide* under the "Transaction-Entered Commands" topic. If a restricted command is issued, you will receive message EVI501I with return code 110 (if rejected by the IMSCMD routine) or message EVI152 (if rejected by IMS).
5. If IMS is customized to restrict the commands the IMS Automation BMP program can issue, message EVI152 will be received for any restricted commands.
6. The IMSCMD routine performs input validation and security checking before passing the command to IMS. If an error is detected, message EVI501 will be issued with a return code and the return code will be passed back to any calling routine. The possible return codes are:

   0 Passed edits. Waiting for PPI processing or IMS command processing messages.

   1-100 Return code from IMSQRY. Refer to return codes for IMSQRY.

103 Input error. The IMS name is missing, the command is missing, or both the IMS name and the command are missing.

104 The name entered is not an IMS subsystem.

105 Security check failure. Operator is not authorized for the specific IMS command.

106 Security check failure. Operator is not authorized for the IMS subsystem.

107 Security check failure. Operator is not authorized to issue any IMSCMD functions.

110 The specific IMS command is not supported thru the IMS automated operator function.

111 Security check failure, unexpected error condition.

112 Security check failure, unexpected return code.

Examples of Usage

Example 1

If you issue:

```
IMSCMD IMS10AA /DIS A
```

IMS Automation will respond:

```
 U AOF01    EVI120I COMMAND ACCEPTED FOR IMS10AA , APPLID = IMS10AA
 = AOF01
EVI690I PPI RESPONSE FROM IMS10AA  FOR FUNCTION IMSCMD   083 00655
EVI691I  ** SUBSYSTEM NAME **
EVI691I        IAOC
EVI691I  REGID JOBNAME   TYPE  TRAN/STEP PROGRAM  STATUS      CLASS
EVI691I      3 MSGIMSAB  TP                       WAITING        5, 6, 7,
EVI691I      2 MSGIMSAA  TP                       WAITING        1, 2, 3,
EVI691I         1 PPIIMSA   BMP   PPIIMSA   EVISPPII
EVI691I        FPRGN     FP    NONE
EVI691I  REGID JOBNAME   TYPE  TRAN/STEP PROGRAM  STATUS
EVI691I        DBRIMSAA  DBRC
EVI691I        DLIIMSAA  DLS
EVI691I  VTAM ACB OPEN          -LOGONS ENABLED
EVI691I  LINE ACTIVE-IN -    1 ACTIV-OUT -    0
EVI691I  NODE ACTIVE-IN -    0 ACTIV-OUT -    4
EVI691I  LINK ACTIVE-IN -    0 ACTIV-OUT -    0
EVI691I  *91154/111742*  IAOC      ACTIVE
EVI692I END
```

Example 2

If you issue:

```
IMSCMD IMS10AA /DIC A (error)
```

IMS Automation will respond:

```
 U AOF01    EVI120I COMMAND ACCEPTED FOR IMS10AA , APPLID = IMS10AA
 = AOF01
EVI690I PPI RESPONSE FROM IMS10AA  FOR FUNCTION IMSCMD   083 00045
EVI691I DFS107 11:22:36 REQUIRED KEYWORD NOT PRESENT
EVI692I END
```

IMSFWM—Add messages to SDF

This routine adds messages to SDF critical message panels.

Note: The IMSFWM common routine will not work unless you code the following control file entry:

```
IMS CRITMSGS,CODE=(*,,,SAVE)
```

See "IMS CRITMSGS—Display critical messages in the SDF" on page 90 for further information.

```
IMSFWM msgtext
IMSFWM TYPE=t,msgtext
```

Keyword and Parameter Definitions

msgtext
The message text and message identifier passed to the Status Display Facility critical message panel.

TYPE=*t*
A 1-character value corresponding to an SDF CRITMSG type entry in the control file. A, E, I, and W are supplied already. Other values may be specified, provided a SDF CRITMSG*t* in the control file corresponds to that type value.

Comments and Usage Notes

This must be called from the automation table because it uses the jobname associated with the message.

If the TYPE=*t* parameter is not specified, *t* is set to the last character of the message ID, and the search is made. If no CRITMSG*t* entry is found, the CRITMSG value will be used.

Examples of Usage

Example 1

```
IF  MSGID='DFS0414I' & TOKEN(2) = 'PERMANENT'
    & TEXT=MESSAGE
    THEN EXEC(CMD('IMSFWM ' MESSAGE) ROUTE(ONE *)) ;
```

Example 2

```
IF  MSGID='DFS3257I' & TEXT=MESSAGE
   THEN EXEC(CMD('IMSFWM 'MESSAGE) ROUTE(ALL *));
```

Example 3

If you wish to see certain IRLM messages in blue reverse video, add an entry like this to the control file:

```
SDF CRITMSGU,CO=B, PR=500,HL=R
```

and call IMSFWM from the message table as follows:

```
IF  MSGID='DXR002I' | MSGID = 'DXR007E' |
    MSGID='DXR011I' | MSGID = 'DXR018E' |
    MSGID='DXR030I' | MSGID = 'DXR034I' |
    MSGID='DXR035I' | MSGID = 'DXR045W' |
    MSGID='DFS626I')
    & TEXT=MESSAGE
   THEN EXEC(CMD('IMSFWM TYPE=U'MESSAGE) ROUTE(ALL *));
          DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
```

Note: A comma is required as a delimiter after the type=U portion of the command.

IMSPOST—Post an external event

Use this routine to set trigger conditions in the status file. See "Defining Startup and Shutdown Triggers" on page 32 for related information.

```
IMSPOST NAME=subsys,FUNCTION={SET|UNSET},EVENT=event
        [,TYPE={STARTUP|SHUTDOWN}]
```

Keyword and Parameter Definitions

NAME=*subsys*
Is used to define the name by which this IMS subsystem is known to SA OS/390.

FUNCTION=
Indicates whether to set the trigger on (SET) or off (UNSET). If SET is specified, then STARTUP or SHUTDOWN must be specified with the TYPE=keyword.

EVENT=*event*
The name of the external condition. This name is used in the EXTCOND and TRIGGER control file entries.

TYPE=
Specifies whether this event is in a STARTUP or SHUTDOWN trigger. This keyword is required if FUNCTION=SET.

Comments and Usage Notes

1. The return codes are:

RC	Meaning
0	Function performed (normal exit)
1	Security check failed
2	Bad keyword
3	IMSQRY (lookup) failed
4	Status file update failed
5	Error on call to EVIEI111 (startup)
5	Error on call to EVIET111 (shutdown)
>**10**	Return code from EVIEI111 or EVIET111

2. The operator submitting the request must be authorized to post external events on this subsystem. After the operator is validated, a lookup routine is invoked to determine the domain ID in which the subsystem resides. If the subsystem resides in the local domain, a routine is called to post the trigger in the local status file. If the subsystem resides on a remote domain, a command is sent to that domain to post the trigger in that domain's status file.

Examples of Usage

Example

```
IMSPOST  NAME=IMS10AA,FUNCTION=SET,EVENT=JOB1DONE,
         TYPE=STARTUP
```

If you code the call in a command list or EXEC, IMS Automation sets the event JOB1DONE on and checks if a startup could be performed.

IMSQRY—Name lookup

Use this routine to retrieve IMS subsystem information.

If a subsystem is in FALLBACK or MOVED status for a particular domain, IMSQRY with LOCAL scope does not recognize the subsystem since this routine only tracks the instance of the subsystem that is being actively managed. An IMSQRY with REMOTE or ALL scope may find the named subsystem elsewhere.

```
IMSQRY REQ={VALIDATE|DETLLIST|MSTRLIST|GET}
       [,NAME={subsystem|group|domain|jobname}]
       [,TYPE={IMS|GROUP|DOMAIN|ANY|JOBNAME}]
       [,SCOPE={LOCAL|REMOTE|ALL}]
```

Keyword and Parameter Definitions

REQ= The request type. The request types are:

VALIDATE
IMS Automation searches for the name (NAME=) and type (TYPE=) specified to validate the name. SCOPE= is only valid with this request if TYPE=IMS.

DETLLIST
IMS Automation searches for the name (NAME=) and type (TYPE=) specified to build a list of subsystems by group or domain. Only TYPE=GROUP or TYPE=DOMAIN are used with DETLLIST. SCOPE= is not valid with the request.

MSTRLIST
IMS Automation searches for the type specified (TYPE=) to build a master list of groups or domains. Only TYPE=GROUP or TYPE=DOMAIN are used with MSTRLIST. SCOPE= is not valid with this request.

GET
IMS Automation searches for a specific IMS subsystem to retrieve the subsystem characteristics. NAME=, and SCOPE= are valid with this request.

NAME={*subsystem|group|domain|jobname***}**
Used with VALIDATE and DETLLIST to provide a specific subsystem, group, domain name or jobname for the search. Used with GET to provide a specific subsystem value. Valid values for the NAME= variables are:

subsystem The name by which an IMS subsystem is known to SA OS/390

group A group of IMS subsystems as defined to IMS Automation

domain The NetView domain as it is coded with NCCFID parameter in DSIDMN.

jobname The jobname by which an IMS subsystem is known to SA OS/390.

TYPE=
Used to provide a specific type. The types are:

IMS
Search for a specific IMS subsystem name, as it is known to SA OS/390. This is the default type. IMS is used with VALIDATE.

ANY
Search for an IMS name first, then a domain, then a group name. If the name is longer than 5 characters the search for a domain is bypassed. ANY is used with VALIDATE.

DOMAIN
The NetView domain name coded in DSIDMN with the NCCFID parameter.

GROUP
The IMS subsystem group name as it is known to IMS Automation.

JOBNAME
Used with GET to provide a specific jobname. Works only when SCOPE=LOCAL and NAME=jobname.

SCOPE=
The scope of systems searched. SCOPE= is used with the VALIDATE and GET requests. It can only be used with VALIDATE when TYPE=IMS. Valid scopes are:

LOCAL
Only the local system is searched for this IMS subsystem.

REMOTE
Only the focal point system is searched for this IMS subsystem.

ALL
First, the local system is searched for this IMS subsystem. If the subsystem is not found on the local system, the request is forwarded to the focal point system.

Comments and Usage Notes

1. Valid IMSQRY return codes include:

RC	Meaning
0	Good
4	An internal error occurred
8	A timeout occurred on a request forwarded to a remote system
12	An internal error occurred
16	The scope is invalid (SCOPE=) for this request
20	A subsystem, group, or domain was not found for the search criteria specified
24	The parameters for this request are invalid
28	An internal error occurred

2. The following are set in the caller's variable pool:

EVILOOKUP_NAME
Set to the value of the NAME= parameter, otherwise set to null.

EVILOOKUP_TYPE
Set to the value of the TYPE= parameter, unless TYPE=ANY in which case it is set to IMS or DOMAIN or GROUP as appropriate.

EVILOOKUP_DCOUNT
The count of IMS subsystems found in a DOMAIN or GROUP when REQ=DETLLIST

EVILOOKUP_DLIST.*n*
The names of the IMS subsystems found in a DOMAIN or GROUP when REQ=DETLLIST

EVILOOKUP_MCOUNT
The count of DOMAINS or GROUPS found when REQ=MSTRLIST.

EVILOOKUP_MLIST.*n*
The names of the DOMAIN or GROUPS found when REQ=MSTRLIST.

EVILOOKUP_JOBNAME
The jobname associated with the subsystem.

EVILOOKUP_DOMAIN
The NetView domain on which the SA OS/390 managing this subsystem is running.

EVILOOKUP_AUTOOPS
The NetView automated operator that handles automation for this subsystem.

EVILOOKUP_USERVAR
The VTAM USERVAR (or generic application ID) associated with this subsystem. This is set to `'********'` if a VTAM USERVAR is not defined.

EVILOOKUP_SUBTYPE
The value of the subsystem ENVIRON.SUBTYPE.

EVILOOKUP_APPLID
The specific VTAM application ID associated with this subsystem.

Refer to Table 8 on page 177.

Table 8. IMSQRY Input and Output Mapping

Input			Output variables (EVILOOKUP. ...)						
REQ=	**TYPE=**	**NAME=**1	**...NAME**	**...TYPE**	**...MCOUNT**	**...MLIST.*n***	**...DCOUNT**	**...DLIST.*n*** 4	**...APPLID** **...DOMAIN** **...USERVAR** **...JOBNAME** **...AUTOOPS** **...SUBID** **...SUBTYPE**
VALIDATE	IMS	R	√	IMS	—	—	—	—	—
	DOMAIN	R	√	DOMAIN	—	—	—	—	—
	GROUP	R	√	GROUP	—	—	—	—	—
	ANY	R	√	√	—	—	—	—	—
MSTRLIST	DOMAIN	—	√	DOMAIN	√	√	—	—	—
	GROUP	—	√	GROUP	√	√	—	—	—
DETLLIST	DOMAIN	R	√	DOMAIN	—	—	√	√	—
	GROUP	R	√	GROUP	—	—	√	√	—
GET	— (IMS implied)	R	√	IMS	—	—	—	—	√

Notes:

1. R indicates that the parameter is required
2. √ indicates that the variable is set
3. — indicates that the variable is not set
4. ...LIST.*n* variables are set if ...COUNT is > 0

IMSRCMD—Request an IMS function

This common routine is used to perform the requested function (CMD=) on the domain where the named IMS resides, whether local or remote. The calling program does not have to be aware of where the IMS resides. It is particularly useful with single-point-of-control as IMSRCMD first determines the domain in which the subsystem resides before building and issuing the request. It then either calls the requested function if the subsystem is on the local domain, or it forwards the command to the remote domain, thus allowing cross-domain communications.

```
IMSRCMD NAME=subsys,[RESP=YES|NO|ACK,]
        [OPER=operator,]CMD=cmd
```

Keyword and Parameter Definitions

NAME=
The name by which the target IMS subsystem is known to SA OS/390, as defined with the SUBSYSTEM control file entry.

RESP=
Send back a response (YES or NO) or just send an acknowledgement (ACK).

OPER=
The operator, on the target domain, that will execute this command. If this is omitted, the automated operator defined with the IMSCNTL control file entry AUTOOPS keyword is used.

CMD=
The requested function to be performed. This may be delimited by single quotes, double quotes, or slashes.

Comments and Usage Notes

The return codes are:

RC	Meaning
0	Good
4	Subsystem name was not supplied
8	Function to be performed was not supplied
12	Incorrect keyword supplied
16	Incorrect parameter for RESP supplied
20	Subsystem was not found on any domain

IMSSEC—Invoke security checking

Security checking can be used to restrict NetView operators using IMS Automation to:

- Perform functions on an installation defined set of IMS subsystems
- Perform an installation defined set of functions on those subsystems

Operator signon and verification are base NetView functions.

```
IMSSEC NAME=subsys,FUNCTION=function
```

Keyword and Parameter Definitions

NAME=
The name by which this IMS subsystem is known to SA OS/390.

FUNCTION=
The function for which checking is performed. The following are defined IMS Automation functions:

IMSCMD This function determines whether an operator can use any IMSCMD function.

/xxx Any IMSCMD command where xxx are the first 3 characters of the supported IMSCMD command. The supported commands are described in the IMSCMD documentation. For example, you would code:

/DIS Secures the IMSCMD /DISPLAY function
/FOR Secures the IMSCMD /FORMAT function

TRIGGERS Use of the TRIGGER function

STARTUP Starts an IMS

SHUTDOWN Shuts down an IMS

SERVICEP Service period functions

SUPPORT Support functions

EVIPPICT IMS Automation program-to-program interface (PPI) functions

BRO Broadcasts a message

Comments and Usage Notes

1. To control access to a function, a command is defined in DSICMD using the CMDMDL statement, as shown:

```
EVISX001   CMDMDL   MOD=EVISX001,RES=Y,PARSE=Y
           CMDSYN   IMSSEC
```

2. IMSSEC keywords are defined with the following NetView statements:

subsys **KEYCLASS** *opclass* Corresponds to a subsystem
function **VALCLASS** *opclass* Corresponds to a function

The keywords are associated with the functions in DSICMD as shown:

IMS10AA	KEYCLASS 1	
STARTUP	VALCLASS 2	
SHUTDOWN	VALCLASS 2	
SERVICEP	VALCLASS 2	
BRO	VALCLASS 2	
SUPPORT	VALCLASS 2	
EVIPPICT	VALCLASS 2	
IMSCMD	VALCLASS 2	
/CHE	VALCLASS 2	/CHECKPOINT command
/DIS	VALCLASS 2	/DISPLAY command
/FOR	VALCLASS 2	/FORMAT command
=OTHER	VALCLASS 3	

Entries are grouped by subsystem, with the subsystem listed as the first item in the group (KEYCLASS). All function items (VALCLASS) following a subsystem item are associated with that subsystem until another subsystem item (KEYCLASS) is encountered.

About =OTHER

=OTHER is used as the default entry and can be used for either KEYCLASS or VALCLASS. This is very useful for defining default security for unspecified IMS subsystems.

3. EVISX001 return codes are:

   0 Good
   4 Keyword not authorized
   8 Value not authorized
   12 Should not occur type error
   9000 Unexpected RC from DSICES or DSIKVS

Examples of Usage

Example 1

```
IMSSEC NAME=IMS10AA,FUNCTION=SHUTDOWN
```

This call checks that the operator has the authority to issue a SHUTDOWN from the operator interface to subsystem IMS10AA.

Example 2

```
IMSSEC NAME=IMS10AA,FUNCTION=PAYR
```

This is an example of a user-defined call where the function PAYR (a user-defined function) is checked for authorization. The following entry must be in DSICMD:

```
PAYR       VALCLASS opclass
```

EVISROUT—IMS message router

The purpose of this common routine is to distribute messages from the common IMS autotask operator, IMSMSTR, to the unique automation operators for each subsystem.

```
EVISROUT [JOBNAME='jobname'] clist [clist-parms]
```

Keyword and Parameter Definitions

JOBNAME=
The jobname of the job this message was issued. Only required when the jobname in the WTO buffer is incorrect. MVS messages issued on behalf of a job may not always contain the correct jobname. Such messages include IEF403, IEF404, and IEF450.

clist
The command list to execute.

clist-parms
Any parameters required by the command list.

Comments and Usage Notes

EVISROUT can be invoked only from the NetView automation table.

Examples of Usage

Example

The following is used to start and shut down IMS message regions:

```
 IF MSGID = 'DFS551I'  & HDRMTYPE = 'E'
    & DOMAINID = 'AOF01'
    THEN EXEC(CMD('EVISROUT EVIES002 ') ROUTE(ALL *))
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
 IF MSGID = 'DFS552I'  & HDRMTYPE = 'E'
    & DOMAINID = 'AOF01'
    THEN EXEC(CMD('EVISROUT EVIES003 ') ROUTE(ALL *))
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);

 IF MSGID = 'DFS810A'  & HDRMTYPE = '>'
    & TEXT = MESSAGE
    & DOMAINID = 'AOF01'
    THEN EXEC(CMD('EHKESORP ' MESSAGE) ROUTE(ALL *))
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y)
         EXEC(CMD('EVISROUT EVIEI00B ') ROUTE(ALL *))
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
```

Adding Local Applications to the IMS Automation Operator Interface

Option 99, Local Applications, from the IMS Automation main menu, enables you to add your local applications to the IMS Automation interface.

To do this, write a module named EVIEU000 using the usage notes described below. We do not provide this module. However, this module is called when option 99 is selected.

These usage notes assume that you understand how to write a NetView panel handler EXEC. These notes clarify unique functions or conventions used with IMS Automation. For your panel to be logically consistent with the IMS Automation interface, incorporate the following usage notes.

Usage notes:

1. To exit IMS Automation (PF2) or to return to the main menu (PF4) code the following after displaying your panel and accepting the input:

```
WHEN VIEWAID = 'PF2' | VIEWAID = 'PF14' THEN
  DO
    EVI_PF2 = 'YES'
    'GLOBALV PUTT EVI_PF2'
    EXIT 0
  END
    and
WHEN VIEWAID = 'PF4' | VIEWAID = 'PF16' THEN
  DO
    EVI_PF4 = 'YES'
    'GLOBALV PUTT EVI_PF4'
    EXIT 0
  END
```

   When you call a module and you return from that module, you should exit if the called module displays a panel and PF2 or PF4 was pressed. To check for this, code the following after the call.

```
'GLOBALV GETT EVI_PF2'
IF EVI_PF2 = 'YES' THEN
  DO
    EXIT 0
  END
    and
'GLOBALV GETT EVI_PF4'
IF EVI_PF4 = 'YES' THEN
  DO
    EXIT 0
  END
```

2. To handle fastpath:

   - Add the following to the beginning of the program:

     ```
     'SIGNAL ON HALT'
     ```

     Add the following routine:

     ```
     HALT:
         EVI_PF2 = 'YES'
         'GLOBALV PUTT EVI_PF2'
         EXIT 0
     ```

   - Add the following code to support fastpath entered on your panel by the operator:

     ```
     WHEN VIEWAID = 'ENTER' & CMD ¬= ''THEN
        DO
          IF SUBSTR(CMD,1,1) = '=' THEN
             DO
               PARSE VAR CMD '=' REST
               CMD = 'EVIE0000 ' || REST
             END
          'CMD HIGH 'CMD
        END
     ```

     Note: In this code, CMD is the command line on the NetView panel.

   - If you code a menu panel, add the following code to support fastpath entry:

     ```
     'GLOBALV GETT EVI_SELECTION'
     IF EVI_SELECTION ¬= ''
        DO
          PARSE EVI_SELECTION MYSELECTION '.' EVI_SELECTION
         'GLOBALV PUTT EVI_SELECTION'
        END
     ```

3. Support for the name of the IMS subsystem:

   - On entry, or returning from a called routine, to get the IMS subsystem name (if the previous routine had a valid name and saved it) code the following:

     ```
     'GLOBALV GETT EVISELNM'
     MYNAME = EVISELNM
     ```

   - Validate the name using the command processor described in “IMSQRY—Name lookup” on page 174. Following is an example of usage:

     ```
     'IMSQRY REQ=VALIDATE,TYPE=IMS,NAME='MYNAME
     IF RC ¬= 0
        DO
         write your error message
        END
     ELSE
        EVISELNM = MYNAME
        'GLOBALV PUTT EVISELNM'
     ```

Part 3. Planning, Installing, and Customizing the IMS Automation Feature

Chapter 5. Planning . . . 187
Pre-Installation and Planning Steps . . . 187
Designing Your Automation Environment . . . 188

Chapter 6. Installation . . . 189
Step 1: Load the Tape Files . . . 189
Step 2: Edit MPFLSTxx . . . 189
Step 3: Update IEAAPFxx in SYS1.PARMLIB . . . 190
Step 4: Define Subsystem Allocatable Consoles . . . 190
Step 5: Check the Subsystem Name Table (IEFSSNxx) . . . 191

Chapter 7. Merge NetView Related Members . . . 193
Step 1: Add IMS Automation Data Sets to NetView JCL . . . 193
Step 2: Copy IMS Automation Sample Members to the Target Library . . . 194
Step 3: Merge SDF Members . . . 194
Step 4: Include IMS Control File Entries . . . 194
Step 5: Include Required Command Model Statements into DSICMD . . . 194
Step 6: Include and Update the Automation Table . . . 195
Step 7: Include Operator Definitions into DSIOPF . . . 196
Step 8: Add Additional VTAM APPL Statements . . . 196
Step 9: Merge or Include the NetView Profile Data Set . . . 196
Step 10: Include Task Statements into DSIDMN and Update . . . 196

Chapter 8. IMS Automation IMS Definitions . . . 197
Step 1: Modify and Run the IMS SYSGEN . . . 197
Step 2: Define IMS PSB Entries . . . 198
Step 3: Define IMS Security Gen Entries . . . 198
Step 4: Define IMS BMP Procedure . . . 199
Step 5: Create PPI Initialization Member for IMS . . . 199
Step 6: Schedule Each IMS to be Stopped and Restarted . . . 199

Chapter 9. IMS Automation Definitions in NetView . . . 201
Step 1: Modify the SA OS/390 ENVIRON Base Control File Entry . . . 201
Step 2: Add the Automation Operators . . . 202
Step 3: Add the Automation Class to the Notification Operator Definitions . . 202
Step 4: Code the Entries for Control Region . . . 203
Step 5: Code the Entries for DBRC/DLISAS Regions . . . 205
| Step 6: Code the Entries for FDR Region . . . 206
| Step 7: Code the Entries for CQS Region . . . 206
Step 8: Code the Entries for Message Regions . . . 206
Step 9: Code the Entries for OLDS . . . 207
Step 10: Code the Entries for MSC Links . . . 208
Step 11: Code the Entries for RECONs Recovery . . . 208
Step 12: Code the Entries for Transaction and Program Recovery . . . 208
Step 13: IMS Automation Security Checking . . . 209
Step 14: Extended IMS Automation Common Control File Definitions . . . 210
Step 15: Preparing IMS Automation to Manage an IMS XRF System in a Dual-CPC Environment . . . 210
Step 16: XRF Considerations in an ARM Environment . . . 212

Step 17: REXX Environment Table 213
Step 18: Customizing the SDF 214

Chapter 10. Optional Additions to the PPI 217
Step 1: Extend EVISPINM Member, If Required 217
Step 2: Extend EVINTASK, If Required 222

Chapter 11. Testing IMS Automation 225
Startup Test Scenario 226
Message Region Test Scenario 227
Shutdown Test Scenario 228

Chapter 5. Planning

This chapter describes the steps you will take to plan for installation, customization, and implementation of IMS Automation.

You should follow each step in the order given and track your progress; checklists are provided to aid you. You will use many of these lists multiple times, so you may want to make copies of them.

Pre-Installation and Planning Steps

Complete the steps listed here before installing the IMS Automation code. Use this table to track your completed work:

Table 9. Planning Check List

√	Step	Action
	1.	Verify that the SA OS/390 on each NetView domain in which IMS Automation will be installed is operational. This is a very important step. Test these applications according to the testing procedures described in the SA OS/390 base documentation. Do not proceed with the installation until this step is complete.
	2.	If you will use single-point-of-control, verify that the focal point hierarchy is defined according to the descriptions in the SA OS/390 base documentation.
	3.	Complete the “Planning Worksheets” on page 364. See “Designing Your Automation Environment” on page 188.
	4.	Schedule IMS subsystem initializations, NetView initializations, system IPLs, and IMS system generations, according to the directions in the worksheet shown in “System IPLs, NetView Restarts, and IMS Restarts” on page 370.

Designing Your Automation Environment

This section explains how to design and plan for automation. While going through this process, keep the following in mind:

- Each IMS region and its dependent regions are defined to IMS Automation as separate subsystems. A set of control file entries are defined for each IMS region and its dependents. Some are required and some are optional.
- This design takes into consideration interactions between NetView domains.
- Required definitions are done before the optional definitions.

Use these steps to design your automation environment:

1. List all of your IMS regions and their dependents by NetView domain. In this step, you will assign a name to each IMS region and each dependent region. IMS Automation will identify the regions by these names. Use the worksheet in “NetView Domains and IMS Regions” on page 364.
2. Gather basic automation information for each IMS subsystem. Use a separate worksheet for each NetView domain. See “IMS Subsystem Information Sheets” on page 365.
3. List the IDs of the IMS Automation automation operators. Use the worksheet in “Automation Operator Information Sheets” on page 366.

The following steps are optional.

4. If you will group IMS subsystems so that they can be started or shut down from the operator interface in groups, fill out the worksheet in “IMS Subsystem Group Information Sheet” on page 367.

   Note: You must have single-point-of-control defined to be able to start up or shut down subsystems across domains.

5. If you will use service periods, fill out the worksheet in “Service Period Information Sheets” on page 368.

Chapter 6. Installation

Complete the steps listed here. Use this table to track your completed work:

Table 10. Installation Checklist

√	Step	Action
	1.	Load the program files. See "Step 1: Load the Tape Files."
	2.	Add the entries needed for IMS Automation to forward messages to NetView. See "Step 2: Edit MPFLSTxx."
	3.	Update IEAAPF*xx* in SYS1.PARMLIB to provide APF authorization. See "Step 3: Update IEAAPFxx in SYS1.PARMLIB" on page 190.
	4.	Define subsystem allocatable consoles. See "Step 4: Define Subsystem Allocatable Consoles" on page 190.
	5.	Check the subsystem name table to ensure that the NetView SSI is first. See "Step 5: Check the Subsystem Name Table (IEFSSNxx)" on page 191.

Step 1: Load the Tape Files

Using the information in the *Program Directory for AOC/MVS IMS Automation Feature*, load the IMS Automation libraries from the distribution tape (using the SMP utility).

Step 2: Edit MPFLSTxx

Add the following entry in MPFLST*xx* in SYS1.PARMLIB to trap all DFS and EVI prefix messages:

```
DFS*,SUP(NO),AUTO(YES)
DSP*,SUP(NO),AUTO(YES)
EVI*,SUP(NO),AUTO(YES)
IOS071*,SUP(NO),AUTO(YES)
DXR*,SUP(NO),AUTO(YES)
AVM0*,SUP(NO),AUTO(YES)
IEF450*,SUP(NO),AUTO(YES)
```

This IMS Automation requirement forwards these messages to NetView.

Step 3: Update IEAAPFxx in SYS1.PARMLIB

NetView must be APF-authorized. To accomplish this, all the libraries from the //STEPLIB concatenation in the NetView start procedure must be APF-authorized. To be authorized, a library's name must appear in the list of authorized libraries in the APF member (IEAAPF*xx*) of SYS1.PARMLIB. After IEAAPF*xx* is updated, MVS must be re-IPLed.

Ensure the AOCIMS.V1R4M0.SEVIMOD library is APF-authorized (unless you are using dynamic allocation in MVS 4.3 or higher). Also, depending on the version of IMS you are executing, authorize:

- AOCIMS.V1R4M0.SEVIMOD2 for Version 2 of IMS
- AOCIMS.V1R4M0.SEVIMOD3 for Version 3 of IMS
- AOCIMS.V1R4M0.SEVIMOD4 for Version 4 of IMS
- AOCIMS.V1R4M0.SEVIMOD5 for Version 5 of IMS.
- AOCIMS.V1R4M0.SEVIMOD6 for Version 6 of IMS.

Note: Do this authorization only if the AOCIMS.V1R4M0.SEVIMOD library is going to added to the STEPLIB chain of the control region JCL.

You can find more information about APF authorization in *MVS/ESA Initialization and Tuning*.

Step 4: Define Subsystem Allocatable Consoles

The NetView program uses the MVS subsystem interface to allow operators to issue MVS system commands from the operator station task used by NetView operators and from an automation task. For each active task that can issue MVS system operator commands, a subsystem allocatable console is required for NetView.

Each IMS automation operator can use a subsystem allocatable console. Ensure that the number of subsystem allocatable consoles is increased by the number of IMS automation operators.

The following publications contain information on how to define subsystem allocatable consoles for your system:

- *MVS/ESA Installation: System Generation Reference*
- *MVS/ESA Input Output Configuration Program Users Guide and Reference*.

Step 5: Check the Subsystem Name Table (IEFSSNxx)

The first active NetView SSI is used for program-to-program interface communication. When a NetView SSI is active and in use by the program-to-program interface (PPI), and another NetView SSI becomes active that is coded higher in the SSN table, then the PPI will switch and use that NetView SSI. If IMS Automation has already signed on to the PPI before the switch occurs, IMS Automation program-to-program communications will be disrupted.

To ensure that disruptions do not occur, do one of the following:

- Make sure that the SA OS/390 SSI entry is the first SSI in the SSN table and the SSI starts during the IPL.
- Use an option available with NetView Version 2 Release 4 or later to specify "NOPPI" on all NetView SSIs except the SSI that IMS Automation uses. This "NOPPI" option is specified as a startup parameter on the SSI JCL.
- If you do not code the SSI that IMS Automation uses in the highest position in the SSN table and you do not use the NOPPI option, then the SSI that is first in the SSN table must be up before IMS Automation initialization and must remain uninterrupted until final termination of IMS Automation.

Check the subsystem name table in MVS SYS1.PARMLIB, member IEFSSN*xx* to verify the NetView SSI used by IMS Automation is first in the list (ahead of all other NetView subsystem names).

Chapter 7. Merge NetView Related Members

This section describes how to build the IMS Automation parameter data sets and assemble the code that enables IMS Automation to operate in the NetView environment.

Use the following table to track your completed work:

√	Page	Action
	193	Step 1: Add IMS Automation Data Sets to NetView JCL
	194	Step 2: Copy IMS Automation Sample Members to the Target Library
	194	Step 3: Merge SDF Members
	194	Step 4: Include IMS Control File Entries
	194	Step 5: Include Required Command Model Statements into DSICMD
	195	Step 6: Include and Update the Automation Table
	196	Step 7: Include Operator Definitions into DSIOPF
	196	Step 8: Add Additional VTAM APPL Statements
	196	Step 9: Merge or Include the NetView Profile Data Set
	196	Step 10: Include Task Statements into DSIDMN and Update

Step 1: Add IMS Automation Data Sets to NetView JCL

Add the following libraries to your NetView JCL procedure. Review the NetView JCL procedure to verify that the IMS Automation data set block size does not cause a problem in the concatenation. Remember that you should specify a block size equal to or larger than that of the largest data set in the concatenation chain in the DCB parameter on the DD statement (DCB=BLKSIZE=...).

Note: For optimum performance, it is recommended that these data sets be high in the concatenation chain.

DD	DSN	Description
DSICLD	AOCIMS.V1R4M0.SEVINCL1	Command lists
CNMPNL1	AOCIMS.V1R4M0.SEVINPN1	Panels
CNMPNL1	AOCIMS.V1R4M0.SEVINPN2	Message online help
DSIMSG	AOCIMS.V1R4M0.SEVINMSG	Communication messages
STEPLIB	AOCIMS.V1R4M0.SEVIMOD	Authorized NetView user link library containing the linked IMS Automation modules.

Step 2: Copy IMS Automation Sample Members to the Target Library

Do one of the following:

- Run job EVISJ010 to copy from the sample library (AOCIMS.V1R4M0.SEVISAMP) those sample members that are not merged with existing members. Tailor the JCL to reflect your operational DSIPARM data set.
- Concatenate the sample library to the DSIPARM data set in the NetView Startup procdedure.

Step 3: Merge SDF Members

To merge these members:

1. Merge EVITREE from the sample library (AOCIMS.V1R4M0.SEVISAMP) into the existing AOFTREE tree structure member. EVITREE contains the tree structure for the IMS Automation SDF panels.

   Note: If you are using the %INCLUDE facility for multiple tree structures, EVITREE must be copied into each.

2. Merge EVIPNLS from the sample library (AOCIMS.V1R4M0.SEVISAMP) into the existing SDF panel definition member. EVIPNLS contains a list of %INCLUDE statements for all of the SDF panels used with IMS Automation.

Step 4: Include IMS Control File Entries

%INCLUDE EVIS2CFG from the IMS Automation sample library (AOCIMS.V1R4M0.SEVISAMP) into the running SA OS/390 control file. (This member was already copied from the IMS Automation sample library in Step 2.) This member contains the %INCLUDE statements that cause IMS Automation control file members to be processed by SA OS/390.

Step 5: Include Required Command Model Statements into DSICMD

%INCLUDE EVICMD from the IMSAO sample library (AOCIMS.V1R4M0.SEVISAMP) into DSICMD before the END statement. (This member was already copied from the IMS Automation sample library in Step 2.) EVICMD contains the additional DSIPARM command model statements required for IMS Automation.

Step 6: Include and Update the Automation Table

To update automation table DSIMSG00:

__ 1. %INCLUDE IMSMSG00 into your NetView/SA OS/390 message table.

IMSMSG00 must be merged with the SA OS/390 message table that is loaded early during NetView initialization. The SA OS/390 sample is AOFMSG00. To accomplish this task, insert the statement:

```
%INCLUDE IMSMSG00
```

after the %INCLUDE for AOFMSGSY in member AOFMSG00. Please note that IMSMSG00 contains %INCLUDE statements for a number of other members that include:

- EVIMCON0 — messages that may be in conflict with other entries in your NetView message table. Please review those messages to verify that there are no conflicts. Document and resolve any conflicts.
- EVIMEVI0 — EVI messages that are used by IMS automation

__ 2. Merge IMSMSG01 into your NetView/SA OS/390 message table.

IMSMSG01 must be merged with the SA OS/390 message table that is used during steady-state NetView operation. The SA OS/390 sample is AOFMSG01. To accomplish this task, insert the statement

```
%INCLUDE IMSMSG01
```

after the %INCLUDE for AOFMSGSY in member AOFMSG01. Please note that IMSMSG01 contains %INCLUDE statements for a number of other members that include:

- EVIMCON1 — messages that may be in conflict with other entries in your NetView message table. Please review those messages to verify that there are no conflicts. Document and resolve any conflicts.
- EVIMDFS1 — IMS/ESA messages used by IMS Automation
- EVIMEVI1 — EVI messages that are used by IMS Automation
- EVIMAVM1 — AVM messages that are used by IMS Automation
- EVIMDSP1 — DSP messages that are used by IMS Automation
- EVIMIOS1 — IOS messages that are used by IMS Automation
- EVIMDXR1 — DXR messages that are used by IMS Automation
- EVIMCQS1 — CQS messages that are used by IMS Automation

Step 7: Include Operator Definitions into DSIOPF

To include this member:

__ 1. %INCLUDE member EVIOPF from the IMS Automation sample library into DSIOPF. (This member was already copied from the IMS Automation sample library in Step 2.) EVIOPF defines the IMS Automation automation operators.

__ 2. Verify that the automation operator names are correct according to the worksheet shown in "Automation Operator Information Sheets" on page 366. This worksheet was filled out during the planning phase.

Step 8: Add Additional VTAM APPL Statements

In your VTAM definitions, verify that there are enough VTAM APPL statements to allow all of the operators to be logged on. You should have one statement for each person who is logged on as a NetView operator; do not include automation operators. If you add VTAM APPL statements, be sure to increase the POSPOOL value in DSIDMN appropriately.

Step 9: Merge or Include the NetView Profile Data Set

This is the profile for the automated operators. Do one of the following:

- Copy EVIPRFAO from the IMS Automation sample library (AOCIMS.V1R4M0.SEVISAMP) into your NetView profile (DSIPRF).
- Concatenate the sample library to the DSIPARM data set in the NetView Startup procdedure.

Step 10: Include Task Statements into DSIDMN and Update

%INCLUDE EVIDMN from the IMS Automation sample library into DSIDMN. (This member was already copied from the IMS Automation sample library in Step 2.) The purpose of EVIDMN is to add the EVINTASK PPI initialization member to DSIDMN.

Chapter 8. IMS Automation IMS Definitions

This chapter describes the basic IMS definitions for IMS Automation. Because of the differences between IMS versions and releases, we attempt to provide some definition instructions specific to IMS where we know there are specific issues to be dealt with. You should therefore refer to IMS documentation while performing these steps, especially to the following manuals:

- *IMS/VS or IMS/ESA System Definition Reference*
- *IMS/VS or IMS/ESA Utilities Reference*

If you are using the IMS DBCTL feature for some of your IMS systems, then this chapter does not apply to any of those IMS systems. The DBCTL system does not use the BMP PPI interface. Instead, it routes all of the messages across the MVS console.

IMS 5.1 has changed how the AOI exit is implemented. In the steps below, we have indicated the different steps to be taken for the IMS 5.1 release.

Use the following table to track your steps.

√	Page	Action
	197	Modify and run the IMS SYSGEN
	198	Define IMS PSB entries
	198	Define IMS Security Gen entries
	199	Define IMS MBP procedure
	199	Create PPI initialization member for IMS
	199	Schedule each IMS to be stopped and restarted

Step 1: Modify and Run the IMS SYSGEN

DBCTL regions do not require IMS sysgen information. Therefore this step can be omitted.

__ 1. Add the statements in AOCIMS.V1R4M0.SEVISAMP member EVISI002 into your IMS gen for each IMS.

```
APPLCTN  PSB=EVISPPII,
               PGMTYPE=(BATCH,,1),
               SCHDTYP=SERIAL
TRANSACT CODE=EVITPPI1, PPI TRANSMISSION TRAN
                MSGTYPE=(MULTSEG,NONRESPONSE),
                SCHD=1,
                INQ=(YES,NORECOV),
                SEGSIZE=0,
                SEGNO=0,
                MODE=SNGL
                SERIAL=YES
```

| __ 2. For IMS 5.1 and higher versions, you can skip this substep. IMS 5.1 removed the AOEXIT keyword from the COMM macro. For earlier releases of IMS, ensure that the COMM macro in the gen contains the following statement:

```
AOEXIT=YES, USE THE AOIEXIT
```

__ 3. Make sure that the APPLID parameter on the COMM macro in the GEN contains the proper applid. The APPLID parameter on the COMM macro for IMS must be specified. This parameter must match the one specified in the APPLID parameter statement in the AOCIMS.V1R4M0 control file, when defining a particular IMS control region. If the APPLID is left to default in the IMS GEN, the MVS jobstep name will be used for the APPLID name for this IMS, thus causing the IMS Automation PPI to fail to initialize. A mismatch will occur at initialization time.

| __ 4. This substep is different for different versions of IMS .

| For pre 5.1 versions:
Make sure that the USERLIB specified for the IMS GEN points to the library containing the DFSAOUE0 supplied by IMS Automation: use SEVIMOD2, SEVIMOD3, or SEVIMOD4, depending on your IMS level.

| For 5.1 and higher versions
The DFSAOUEO module no longer needs to be linked in the IMS gen process. It is sufficient to place it in an APF authorized library included in STEPLIB for the IMS region. This could be the RESLIB, or another library in the concatenation.

__ 5. Perform a gen for each IMS.

Step 2: Define IMS PSB Entries

__ 1. Merge the statements in AOCIMS.V1R4M0.SEVISAMP member EVISI001 with your existing PSB gen for each IMS, or manually add the following to it:

```
PCB    TYPE=TP,MODIFY=YES,EXPRESS=YES
PSBGEN LANG=ASSEM,PSBNAME=EVISPPII,CMPAT=YES,IOASIZE=1024
END
```

__ 2. Run the PSB and ACB gens for each IMS.

Note: The output of the ACBGEN utility for the PSB “name” will indicate how much space is required in the CSA PSB pool.

Step 3: Define IMS Security Gen Entries

Add the statements in AOCIMS.V1R4M0.SEVISAMP member EVISI003 to your security maintenance utility input and run a security gen for each IMS, or code them as shown below. This will give transaction EVITPPII access to all commands.

```
)( CTRANS   EVITPPI1    /* GENERATE TRANSACTION SECURITY */
   TCOMMAND *           /* ALL COMMANDS */
```

Note: Security checks will be performed in NetView before IMS Automation is invoked.

Step 4: Define IMS BMP Procedure

The BMP will handle communications between the IMS AOIEXIT and the NetView program-to-program interface. The BMP is initialized as a started procedure requested by EVInAOIX (n=W, X, V, U, or T). Its start is unconditional and not modified by any ACF controls. To customize the BMP, perform the following steps:

__ 1. Copy SEVISAMP member EVISI004 to your user PROCLIB as the name PPI*imsid*, where *imsid* is the one- to four-character IMS ID of the IMS control region.

If you need to change the name of the procedure to start the BMP, you must also change it in the corresponding IMS program-to-program interface initialization member, EVISPINM. See Chapter 10, “Optional Additions to the PPI” on page 217 for details.

__ 2. Update the STEPLIB statement to point to the IMS RESLIB and SEVIMOD*x* you are using.

__ 3. Update the PROCLIB statement to point to the PROCLIB you are using.

__ 4. Update the *imsid* variable to the IMS ID of the control region you are using.

Step 5: Create PPI Initialization Member for IMS

This initialization member, EVISPINM, defines what IMS messages to send to NetView and defines PPI control information.

Note: You can use a single table for every IMS, or, optionally, you can have a separate table for each IMS. The following instructions assume you will use a single table. To create separate tables, see Chapter 10, “Optional Additions to the PPI” on page 217.

Perform the following steps:

__ 1. Use the JCL shown in “EVISJ020” on page 234 to assemble your program-to-program interface initialization member, EVISPINM.

__ 2. Place the assembled member into a library accessible by both the control region and the BMP procedure created in “Step 4: Define IMS BMP Procedure,” typically IMS RESLIB.

Step 6: Schedule Each IMS to be Stopped and Restarted

For each IMS subsystem that will be automated, schedule a time to shut down and restart the subsystem, and recycle each IMS.

Note: Make sure that the CSAPSB parameter that specifies the size of the CSAPSB pool is large enough to handle the new PSB “name.”

Chapter 9. IMS Automation Definitions in NetView

You customize IMS Automation for your specific installation by modifying the control file. Before beginning customization, complete the work sheets provided in “Customization Worksheets” on page 372.

Use the following table to track your completed work:

√	Page	Customize the following definitions:
	201	Step 1: Modify the SA OS/390 ENVIRON Base Control File Entry
	202	Step 2: Add the Automation Operators
	202	Step 3: Add the Automation Class to the Notification Operator Definitions
	203	Step 4: Code the Entries for Control Region
	205	Step 5: Code the Entries for DBRC/DLISAS Regions
	206	Step 8: Code the Entries for Message Regions
	207	Step 9: Code the Entries for OLDS
	208	Step 10: Code the Entries for MSC Links
	208	Step 11: Code the Entries for RECONs Recovery
	208	Step 12: Code the Entries for Transaction and Program Recovery
	209	Step 13: IMS Automation Security Checking
	210	Step 14: Extended IMS Automation Common Control File Definitions
	210	Step 15: Preparing IMS Automation to Manage an IMS XRF System in a Dual-CPC Environment
	213	Step 17: REXX Environment Table
	214	Step 18: Customizing the SDF

You can change many of the SA OS/390 control file entries needed for IMS Automation by using the SA OS/390 customization dialogs. To learn more about the customization dialogs, refer to the SA OS/390 base documentation. For some entries, however, you must manually edit the control file. The examples in this book show how the entries appear when you view the control file.

Step 1: Modify the SA OS/390 ENVIRON Base Control File Entry

For IMS automation to work, you must modify the ENVIRON entry. Add the following EXITS entry to the ENVIRON SETUP control file entry:

```
ENVIRON      SETUP,...,
                   ...,
             EXITS=(EVIEE001,userexit1,userexit2,...)
```

Note: If the EXITS entry already exists, modify it to include EVIEE001 as shown.

You can use the SA OS/390 dialogs to add this entry.

Step 2: Add the Automation Operators

Add the IMS Automation automation operators to the SA OS/390 control file. Code the IMS Automation automation operators as follows:

```
AUTOOPS          IMSMSTR,ID=(AUTIMS),
                 MSG=(DFS*,AVM005*,AVM006*,IOS071*,DSP*,EVI*,DXR*)
AUTOOPS          IMSWATCH,ID=(AUTSURV)
AUTOOPS          IMSPPI,ID=(AUTIPPI)
```

Note: Make sure that these automation operators are defined in the DSIOPF member in the DSIPARM data set of NetView.

You can use the SA OS/390 customization dialogs to add these entries.

Step 3: Add the Automation Class to the Notification Operator Definitions

Update the CLASS parameter on the NTFYOP entries in the control file to reflect the following message class required to support IMS Automation:

60

For example:

```
NTFYOP NETOP1,OPER='NETOPER 1',CLASS=(60)
```

You can use the SA OS/390 customization dialogs to add these entries.

Step 4: Code the Entries for Control Region

For all IMS control regions, code the control file entries shown in Table 11.

Note: Additional entries that are used for specific types of regions and other functions are listed in separate steps.

Table 11 (Page 1 of 3). Applicable Entries for Control Region

√	Required	Control file entry	Comments	See Page
	√	ABCODES	Non-XRF abend codes.	47
	√	ACORESTART	Code as shown with the appropriate subsystem identifier.	51
		ACT CODES	XRF only.	47
		ALT CODES	XRF only.	47
	√	AUTOOPS (Additional)	Ensure that there is an automated operator for each IMS subsystem.	54
		BRO	IMS control regions only. Code to issue the appropriate broadcast message prior to shutdown.	55
		CHE	XRF only.	56
	√	DFS810A	IMS control regions only. Code for response to message DFS810A.	69
	√	DFS989I	DB control regions only. Code for response to message DFS989I.	71
	√	DFS994I	Code for actions after message DFS994I.	73
		DFS3869A DFS3869I	XRF only. Specify exactly as indicated.	66
	√	ENVIRON	None.	76
		EXTCOND	Trigger descriptions.	84
		FORCE	XRF only. Code to inform the old *active* that it is no longer the *active* when a predatory takeover occurs (the *alternate* assumes control).	86
		HOLDQ	Code to hold BMP initiators.	87
	√	IMSCNTL	None.	88
		IMSRSENM	XRF only.	92
	√	INITSTART	Specify any EXITS=(...) you use.	93
		IPS	XRF only. Code this command to adjust MVS processing if necessary during an XRF takeover.	95

Table 11 (Page 2 of 3). Applicable Entries for Control Region				
√	**Required**	**Control file entry**	**Comments**	**See Page**
		POSTCHKP	Code to reflect your installation's procedure to issue commands after a shutdown checkpoint has been issued.	100
		PRECHKP	Code to reflect your installation's procedure to issue commands prior to a shutdown checkpoint being issued.	101
		RELEASEQ	Code to release BMP initiators.	106
	√	RESTART	Specify any EXITS=(...) you use.	107
		RESTARTABORT	Code to reflect desired response to the receipt of messages DFS0618 and DFS166I.	108
		SERVICE	Service period definitions	112
	√	SHUTFORCE	None.	115
	√	SHUTIMMED	None.	115
	√	SHUTNORM	None.	115
	√	SHUTTYPES	None.	120
		SNAPQ	XRF only. Code to reflect your installation's procedure to issue the /SNAPQ command.	123
		START*n*	Assuming that you have coded STARTCMD=YES for the region, this is an optional entry. Code START1 (and START2 for XRF partners, if any) to use start commands other than the standard MVS start command.	124
	√	STARTUP	Code as shown for control region.	126
	√	STOPBMPREGION	None.	128
	√	STOPFPREGION	IMS control regions only.	129
	√	STOPREGION	IMS control regions only.	130
	√	SUBSYSTEM	Code all parameters except STARTOPTIONS and SHUTOPTIONS.	133
		TCO	None.	136
		TCOMEMBERS	None.	137
	√	THRESHOLDS	None.	139
	√	TPABEND	Code exactly as shown with the appropriate subsystem identifier.	141
		TRIGGERS	Trigger definitions.	142
		UNLKAVM	XRF only. Code exactly as shown with the appropriate subsystem identifier.	144

√	Required	Control file entry	Comments	See Page
		UNLOCK	XRF only. Code to reflect your installation's procedure to issue the /UNLOCK command.	145
		USERSTART	User-defined unique IMS restart commands	147
		VTAMTERMS	IMS control regions only. This entry is used in both an XRF IMS and a non-XRF IMS environment.	148
		$PI	XRF only. Code to reflect the draining of the initiator scheme for the IMS subsystem in an XRF environment.	150
		$SI	XRF only. Code to reflect the startup of the initiators after an XRF takeover.	151
		$TI	XRF only. Code to reflect the setup of the initiator scheme for the IMS subsystem in an XRF environment.	152

Table 11 (Page 3 of 3). Applicable Entries for Control Region

Step 5: Code the Entries for DBRC/DLISAS Regions

For DBRC/DLISAS regions, code the control file entries shown in Table 12.

Table 12. Applicable Entries for DBRC/DLISAS Eegions

√	Required	Control file entry	Comments	See Page
	√	ENVIRON	Code SUBTYPE of either DLS for DLISAS region or DBRC for DBRC region.	76
	√	SUBSYSTEM	Code all subsystem parameters including STARTOPTIONS=PARENT and SHUTOPTIONS=PARENT.	133
	√	THRESHOLDS	None.	139

Step 6: Code the Entries for FDR Region

For FDR regions, code the control file entries shown in Table 13.

Table 13. Applicable Entries for FDR Regions

√	Required	Control file entry	Comments	See Page
	√	SHUTNORM SHUTIMMED	None	115
	√	SUBSYSTEM	None	133

Step 7: Code the Entries for CQS Region

For CQS regions, code the control file entries shown in Table 14.

Table 14. Applicable Entries for CQS Regions

√	Required	Control file entry	Comments	See Page
	√	SHUTNORM SHUTIMMED	None	115
	√	SUBSYSTEM	None	133

Step 8: Code the Entries for Message Regions

For the message regions, code the control file entries shown in Table 15.

Table 15 (Page 1 of 2). Applicable Entries for Message Regions

√	Required	Control file entry	Comments	See Page
	√	ENVIRON	Code SUBTYPE of TP and the appropriate installation-defined SYNC parameter value. (SYNC is used for XRF only).	76
		SHUTNORM SHUTIMMED SHUTFORCE	These entries are required if you do **not** code SHUTOPTIONS=PARENT.	115
		START*n*	If you have coded STARTOPTIONS=PARENT for the region, you must code START*n* or IMS Automation will not start the region. Assuming that you have coded STARTCMD=YES for the region, this is an optional entry. Code START1 (and START2 for XRF partners, if any) only if your STARTUP entry (126) specifies "EVIEI00T."	124
		STARTUP	This entry is required if you do **not** code STARTOPTIONS=PARENT.	126

Table 15 (Page 2 of 2). Applicable Entries for Message Regions				
√	**Required**	**Control file entry**	**Comments**	**See Page**

	√	SUBSYSTEM	Code all subsystem parameters. Code either:
			`STARTOPTIONS=PARENT`
			`SHUTOPTIONS=PARENT`
			or
			`STARTCMD=YES`
				133

	√	THRESHOLDS	None.	139

Step 9: Code the Entries for OLDS

For IMS recovery components, code the control file entries shown in Table 16.

Table 16. Applicable Entries for Online Data Sets (OLDS)				
√	**Required**	**Control file entry**	**Comments**	**See Page**
	√	OLDS	Code the names of OLDS to be kept as spares. IMS Automation starts the spares only when the number of available OLDS drops below the minimum needed.	97
		RECOVERY	Code the RECOVERY *subsys.olds* entry with a value of AUTO=Y.	104
	√	THRESHOLDS	Code the THRESHOLDS *subsys.olds* entry to set the critical, infrequent, and frequent error thresholds for OLDS.	139
	√	DFS3258A	Action to take on last OLDS.	65

Step 10: Code the Entries for MSC Links

For MSC links, code the control file entries shown in Table 17.

Table 17. Applicable Entries for MSC Links

√	Required	Control file entry	Comments	See Page
	√	DFS2142	Code this entry to restart a logical link path.	61
	√	DFS2161I	Code this entry to restart a link after it has been stopped by an IMS system.	63
	√	DFS2169I	Code this entry to restart a link after disconnection of a Multiple Systems Coupling (MSC) link between two IMS systems.	64
		RECOVERY	Code with value of AUTO=Y.	104
	√	THRESHOLDS	None.	139

Step 11: Code the Entries for RECONs Recovery

For RECON recovery, code the control file entries shown in Table 18.

Table 18. Applicable Entries for RECONs

√	Required	Control file entry	Comments	See Page
		RECONS	To turn on active monitoring for RECONs, code this entry. IMS Automation checks for spare RECONs at the interval you specify on the RECONS entry.	103

Step 12: Code the Entries for Transaction and Program Recovery

For transaction recovery, code the control file entries shown in Table 19.

Table 19. Applicable Entries for Transaction and Program Recovery

√	Required	Control file entry	Comments	See Page
	√	ABCODEPROG	Code this entry to specify system action in response to program abend codes.	45
	√	ABCODETRAN	Code this entry to specify system action in response to transaction abend codes.	49
	√	DFS554A	Code this entry to restart a transaction and program after an abend.	67
		RECOVERY	Code with value of AUTO=Y.	104
	√	THRESHOLDS	None.	139

Step 13: IMS Automation Security Checking

Edit DSICMD and define, by class, those operators authorized to perform specific functions against specific IMS subsystems. The operator classes are coded KEYCLASS and VALCLASS. KEYCLASS can be a IMS subsystem name (as defined to IMS Automation), a domain, or a group name (also as defined to IMS Automation). VALCLASS identifies functions allowed for the subsystem, group, or domain specified with the previous KEYCLASS entry. All IMS subsystems must have security definitions in DSICMD. The sample member EVISECUR is provided for that purpose. Table 20 shows sample security definitions.

Table 20. Security Definitions for IMS Subsystems

Definition	Description of function
`IMS100AA    KEYCLASS 1`	Class 1 operators can work with IMS10AA.
`INQUIRY    VALCLASS 1`	Class 1 operators can invoke an inquiry on IMS10AA.
`STARTUP    VALCLASS 2`	Class 2 operators can invoke a startup on IMS10AA.
`SHUTDOWN   VALCLASS 2`	Class 2 operators can invoke a shutdown on IMS10AA.
`SERVICEP   VALCLASS 2`	Class 2 operators can work with service periods on IMS10AA.
`TRIGGERS   VALCLASS 2`	Class 2 operators can work with triggers on IMS10AA.
`BRO        VALCLASS 2`	Class 2 operators can broadcast messages IMS10AA
`SUPPORT    VALCLASS 2`	Class 2 operators can work with support function on IMS10AA.
`EVIPPICT   VALCLASS 2`	Class 2 operators can start and stop the PPI on IMS10AA.
`=OTHER     VALCLASS 3`	Class 3 operators can perform any other function on IMS10AA not already defined in this group of netries (default).
`=OTHER     KEYCLASS 9`	Class 9 operators can work with any other IMS subsystem.
`=OTHER     VALCLASS 9`	Class 9 operators can perform any function on any other IMS subsystem.

Notes:

1. =OTHER is used as the default entry and can be used for either KEYCLASS or VALCLASS.

You can use IMS Automation security checking for your own functions using the same framework described here. Security checking is called by the routine described in “IMSSEC—Invoke security checking” on page 179.

Step 14: Extended IMS Automation Common Control File Definitions

If required, define the following control file entries:

__ 1. Use the worksheet shown in "IMS Subsystem Group Information Sheet" on page 367 to code the control file entry described in "IMSGROUP—Group of IMS subsystems" on page 91.

__ 2. If you will customize state/action tables, code the control file entries described in "AREA—Define a set of state/action tables" on page 52 and "PRODUCT—Subsystem state/action table sets" on page 102.

Step 15: Preparing IMS Automation to Manage an IMS XRF System in a Dual-CPC Environment

Perform the following step only if you are running an IMS subsystem that is XRF and using shared DASD within a dual-CPC environment. If your IMS configuration is XRF and the ACTIVE and ALTERNATE subsystems execute on separate CPCs then you must perform the following steps. An example for each of these steps follows.

__ 1. Define additional TASK statements in the DSIDMN member, which is located in DSIPARM.

__ 2. Define DSTINIT members, one for each DOMAIN, in the NetView DSIPARM data set. Use EVISTSM example in the SEVISAMP data set.

__ 3. Update the IMS Automation control file with STSFILE entries.

__ 4. Verify that SA OS/390 status file placement is in VSAM user catalogs that are accessible from the master catalogs of both CPCs.

Coding example

The following is an example for two NetView domains managing an IMS XRF system in a dual-CPC environment. The first domain is DOM01. The second domain is DOM02.

1. Define additional TASK statements in the DSIDMN member, located in DSIPARM. In domain DOM01 define the task that reads the status file for DOM02.

   ```
   TASK   MOD=DSIZDST,TSKID=AOFDOM02,MEM=STSDOM02,PRI=6,INIT=N
   ```

   In domain DOM02, define the task that will read the status file for DOM01.

   ```
   TASK   MOD=DSIZDST,TSKID=AOFDOM01,MEM=STSDOM01,PRI=6,INIT=N
   ```

2. Define DSTINIT members, one for each DOMAIN, in the NetView DSIPARM data set. When this is done, change the DSTINIT PDDNM value to match the unique DD name formed by concatenating "AOF" with the domain name. Use EVISTSM example in the SEVISAMP data set.

   In domain DOM01, define the DSTINIT member called STSDOM02. The member name would be STSDOM02.

```
***********************************************
* DSTINIT MEMBER FOR IMS Automation XRF STATUS FILE     *
* DEFINES STATUS FILE IN OTHER DOMAIN         *
***********************************************
* COMMENT LINE
  DSTINIT PDDNM=AOFdom02
  DSTINIT XITVN=AOFSISTS
  DSTINIT FUNCT=VSAM
  DSTINIT DSRBO=1
  * END OF MEMBER
```

   In domain DOM02, define the DSTINIT member called STSDOM01.

```
***********************************************
* DSTINIT MEMBER FOR IMS Automation XRF STATUS FILE     *
* DEFINES STATUS FILE IN OTHER DOMAIN         *
***********************************************
* COMMENT LINE
 DSTINIT PDDNM=AOFdom01
 DSTINIT XITVN=AOFSISTS
 DSTINIT FUNCT=VSAM
 DSTINIT DSRBO=1
* END OF MEMBER
```

   Ensure that the DSTINIT parameters start in the second column; this is a NetView requirement.

3. Update the IMS Automation control file with STSFILE entries. The update to the IMS Automation control file is in both domains. The entry in domain DOM01 would appear like this:

```
STSFILE  DOM02,DSN=AOFdom02.DOM02.STATS
```

   The entry in domain DOM02 would appear like this:

```
STSFILE  DOM01,DSN=AOFdom01.DOM01.STATS
```

   The DSN name is the fully qualified name for the SA OS/390 status file.

4. The SA OS/390 sample AOFJS007 specifies the DEFINE CLUSTER control statement used to establish the SA OS/390 status file. The SHR(2) operand needs to be changed to SHR(3) and the status file reallocated in order to permit the sharing described in this step. Also the new allocation must be on a shared volume accessible by both system images in the dual-CPC environment.

5. Use the VSAM ALIAS command to ensure that both status files are in USER catalogs that are accessible from each system's MASTER catalog.

Step 16: XRF Considerations in an ARM Environment

The dual-CPC environment discussed in “Step 15: Preparing IMS Automation to Manage an IMS XRF System in a Dual-CPC Environment” on page 210 must be scaled up somewhat in order to fully recover an IMS XRF-alternate subsystem to a secondary system.

IMS automation is:

- Based upon the IMS design for coordinating XRF and ARM methods of recovery. When ARM recovery is used, ARM always recovers only the XRF-alternate subsystem. The XRF-active is recovered through XRF processing.
- The XRF-alternate is recovered by ARM either by restarting it on the same system image or by moving it to a secondary system.
- The original XRF-active is never moved. When failure does occur and the XRF-alternate assumes active status, the original XRF-active may be recovered and assume the role of XRF-alternate, remaining on the same system image. In this role, the “new” XRF-alternate is not ARM-recoverable to any other system image except that it may be restarted in its own system image; that is, a restart in-place is allowed. In other words, the original XRF-active has no secondary system-image locations defined. You must enforce this by not defining secondary locations for the original XRF-active subsystem.

  Without this restriction, a series of ARM/XRF-recovery cycles has the potential to “walk” the XRF pair all through the set of possible secondary systems, creating coordination and management problems. The design of the IMS automation feature has been done in this manner so as to maximize recoverability while minimizing your overhead regarding system installation, customization, and test.
- To complete the description, in a non-XRF environment, it is possible for an IMS subsystem (i.e. the control region) to be moved to any system image that has been configured to be a secondary location.

Each potential XRF-alternate subsystem located at a secondary system image must have the following automation-control file (ACF) settings established in order to assume the role as an XRF-alternate if and when called upon. Unless specifically stated, these apply solely to the control region (e.g. ENVIRON.SUBTYPE=CTL):

- An STSFILE statement must exist which has the name of the NetView domain of the original XRF-active subsystem, and the dataset name of the SA OS/390 status file of that same domain.
- The value of the ARMNAME= keyword of the SUBSYSTEM statement must be set to the same value as that of the original XRF-alternate.
- The value of the SECONDARY= keyword of the ASSOCIATION statement must include the secondary-system identifier as one of the entries in the list.
- The value of the PRTNRDOM= keyword of the ENVIRON statement must be set to the name of the NetView domain containing the XRF-active subsystem.
- The value of the PRTNRID= keyword of the ENVIRON statement must be set to the ID of the XRF-active subsystem.

- The value of the PRTNRSUB= keyword of the ENVIRON statement must be set to the subsystem name of the XRF-active subsystem. This holds for all region subtypes that would normally cross-reference each other in this manner.

Step 17: REXX Environment Table

Refer to *AOC/MVS Planning and Installation* to review the number of REXX environments

Step 18: Customizing the SDF

Note: Before using these steps, refer to the SA OS/390 base documentation for a complete overview on customizing the SDF. Also read "Setting Up SDF for IMS Automation" on page 25.

Note: The SDF members were merged in "Step 2: Copy IMS Automation Sample Members to the Target Library" on page 194 and "Step 3: Merge SDF Members" on page 194.

__ 1. Depending on how you are currently running your system, you can use the SDF panels in one of two ways. If you are the generic entry, SUBSYS, where each individual system is not specifically defined in the tree, such as:

```
2 SYSTEM
  3 SUBSYS
2 IMS
  3 IMSMSG
  3 IMSARCH
  3 IMSMSCL
  3 IMSOLDS
  3 IMSRECN
  3 IMSTIMR
  3 IMSTRAN
```

Then the supplied panels will all be used to display the information related to IMS.

If you are running without SUBSYS (SA OS/390), such as:

```
2 SYSTEM
  3 JES
  3 VTAM
  3 IMS1
  3 IMS2
  .
  .
  .
2 IMS
  3 IMSMSG
  3 IMSARCH
  3 IMSMSCL
  3 IMSOLDS
  3 IMSRECN
  3 IMSTIMR
  3 IMSTRAN
```

with each IMS system specifically defined in the tree, then all the IMS updates except for IMSMSG will be placed on the specific system entry for that IMS when the alerts occur.

Depending on which method you choose, add an entry to your primary SDF panel definition member (the first panel displayed when either the SDF command is used) as shown:

If you are not using SUBSYS (SA OS/390)

```
SF(SY1.IMSMSG,05,72,73,N, ,EVIDMSG1)
ST(I)
```

If you are using SUBSYS (SA OS/390)

```
SF(SY1.IMS,05,72,73,N, ,EVID0001)
ST(I)
```

Note: These examples assume that your system name is SY1.

__ 2. Adjust the row, starting column, and ending column so that the letter I is under the features heading, as shown in the following example:

```
SYSTEM                AOC/MVS SUPPORT SYSTEMS

   System   Subsystems   WTORs     Gateways   Spool        MVS Comps  Features

   DALLAS                                                             I  C  O

   RALEIGH                                                               C

   CHICAGO                                                            I

   DENVER                                                                   O

   PHOENIX  FPIMSEA      IMS401E   AOF05I     JES                     I  C  O
```

__ 3. (Optional step—Do this step if you want the panels to be resident). Add EVIPNLS to your existing AOFPNLS (SA OS/390).

__ 4. If the SDF root name, as defined with the ENVIRON SETUP,SDFROOT= control file entry, is not SY1, you will need to edit the following members and change SY1 to your system name in the SDF status field component name.

______	EVIDARCA	______	EVIDMSG1	______	EVIDTIMA
______	EVIDARCB	______	EVIDMSG2	______	EVIDTIMB
______	EVIDARC1	______	EVIDOLDA	______	EVIDTIM1
______	EVIDARC2	______	EVIDOLDB	______	EVIDTIM2
______	EVIDMSCA	______	EVIDOLD1	______	EVIDTRNA
______	EVIDMSCB	______	EVIDOLD2	______	EVIDTRNB
______	EVIDMSC1	______	EVIDRECA	______	EVIDTRN1
______	EVIDMSC2	______	EVIDRECB	______	EVIDTRN2
______	EVIDMSGA	______	EVIDREC1	______	EVID0001
______	EVIDMSGB	______	EVIDREC2		

__ 5. Customize the panels as required.

__ 6. Refer to “SDF support” on page 110 to code the IMS support definitions as required.

Chapter 10. Optional Additions to the PPI

NetView's program-to-program interface (PPI) provides the ability to communicate between a NetView application and other address spaces on the same host, such as IMS. For a detailed description of the NetView PPI, refer to the NetView documentation. The PPI makes it possible for NetView automation to have cooperative execution of commands in IMS and command processors in NetView.

IMS Automation uses the NetView PPI to:

- Initiate the execution of an IMS command from NetView
- Process a response from this IMS command
- Send IMS messages from IMS to NetView that drive resulting command lists or command processors

Both IMS and NetView have IMS Automation program-to-program interface components. The IMS Automation initialization member for IMS is EVISPINM. On the NetView side, EVINTASK is the initialization member. You can fully customize EVISPINM and, in some cases, you can change EVINTASK. Both PPI components use the ENQ/DEQ mechanism for conversation. This can be observed by an exclusive enqueue on the relevant resource involved. This is a normal situation for an AOC/MVS-IMS PPI conversation and can be ignored.

This chapter describes the PPI initialization members, EVISPINM and EVINTASK, and explains how to change them to meet special system needs. The steps in this chapter are optional. Use the following table to track your steps.

√	Page	Action
	217	Step 1: Extend EVISPINM Member, If Required
	222	Step 2: Extend EVINTASK, If Required

Step 1: Extend EVISPINM Member, If Required

EVISPINM is used on the IMS side of the program-to program interface to:

- Describe the search criteria to be used when processing messages in the automation operator exit and to determine whether the messages are to be sent to NetView, sent to the system console, or suppressed.

  | **Note:** Messages specified in EVISPINM affect only MTO messages or the
  | MTO copy of the message if it is also issued as a WTO.

  | Thus, if IMS puts a message out as a WTO, the parameters specified in
  | EVISPINM have no bearing on this. For further information, see the
  | *IMS/ESA Customization Guide*, "Automated Operator Exit Routine
  | (DFSAOUE0)".

- Set the program-to-program interface buffer queue limit. This is the number of outstanding buffers that can be stored in the receiver buffer queue.
- Define the program-to-program interface receiver identifier for the EVISNPPI NetView subtask program.
- Define a unique job name for the IMS Automation BMP job if you do not want to use the default, PPI*imsid*.

- Define the default return code from the automated operator exit.
- Contain the IMS name to generate a unique table for an IMS. You use this entry only if you want to define a separate table for each IMS instead of using one table for all.
- Define the transaction ID that the routine AOIEXIT uses to send messages to the BMP.
- If you do not want to use the default name, EVIAOUE0, define the name of a user-written AOI exit that IMS Automation should call instead.
- Describe the relationship between function names and IMS transaction names.

If any changes are required to the default EVISPINM, do the following:

__ 1. Copy EVISPINM (located in SEVISAMP) into a new member named EVII*ims_id*.

__ 2. Edit EVII*ims_id* and:

   a. Update the IMSID.

   b. Change the RECEIVERID= name, if required. This value must be the same as the value defined in “Step 2: Extend EVINTASK, If Required” on page 222.

   c. Change the PPIPROC name, if required. For example, if you changed the name of the procedure to start the BMP, you must also change the name in the corresponding IMS program-to-program interface initialization member (EVISPINM or EVII*ims_id*).

   d. Add your own entries, if required.

__ 3. Use the JCL shown in “EVISJ020” on page 234 to assemble the program-to-program interface initialization member EVII*ims_id*.

__ 4. Place the assembled member into a library accessible by both the control region and the PPI proc, typically IMS RESLIB.

Figure 17 shows a sample of the information contained in the EVISPINM member.

```
EVIMPINM TYPE=INITIAL,       INITIAL ENTRY
      IMSID=ims_id           MUST MATCH IMSID SPECIFIED
      BUFFQL=15,             BUFFER QUEUE LIMIT
      RECEIVERID=NETVIPPI,   NETVIEW RECEIVER IDENTIFICATION
      DEFAULTRC=CANCEL,      CANCEL (12) OR IGNORE (04)
      TRANID=EVITPPI1        TRANSACTION NAME
      SENDERID=applid        IMS SENDER IDENTIFICATION
      PPIPROC=name           MESSAGE-DRIVEN BMP PROC NAME
      EXITNAME=AIOEXIT       USER-DEFINED AOI EXIT MODULE

EVIMPINM TYPE=SEARCH,MSGID=DFS551I,SUP=NO,AUTO=YES,
EVIMPINM TYPE=SEARCH,MSGID=DFS552I,SUP=NO,AUTO=YES
EVIMPINM TYPE=SEARCH,MSGID=DFS0414I,SUP=NO,AUTO=YES
EVIMPINM TYPE=SEARCH,MSGID=DFS2161I,SUP=NO,AUTO=YES
EVIMPINM TYPE=SEARCH,MSGID=DFS2168I,SUP=NO,AUTO=YES
EVIMPINM TYPE=SEARCH,MSGID=DFS2169I,SUP=NO,AUTO=YES
EVIMPINM TYPE=SEARCH,MSGID=DFS554A,SUP=NO,AUTO=YES
EVIMPINM TYPE=SEARCH,MSGID=DFS994I,SUP=NO,AUTO=YES
EVIMPINM TYPE=SEARCH,MSGID=D,SUP=YES,AUTO=NO

EVIMPINM TYPE=ENTRY,      DEFINE A FUNCTION
      FUNCTION=IMSCMD,    FUNCTION NAME
      TRANSID=IMS         TRANSACTION NAME

EVIMPINM TYPE=FINAL       REQUIRED END
```

Figure 17. EVISPINM PPI Initialization Member

Each TYPE= keyword has a set of parameters you can customize. The parameters in EVISPINM are as follows:

TYPE=INITIAL

Must be the first EVIMPINM specification.

Optionally, the IMS buffer queue limit and the NetView receiver identifier can be specified. The valid parameters for TYPE=INITIAL are:

IMSID=

Specifies the IMS ID of the procedure to be controlled by the processing table.

This option will allow unique processing tables for different IMS control regions in which the control regions do not have unique datasets for every IMS procedure.

If omitted, the name of this load module will have the default name of EVISPINM.

If specified, the name of this load module will be EVII*ims_id*

BUFFQL=

Specifies the IMS buffer queue limit for commands sent from NetView to IMS using the PPI.

If omitted, a default of 3 is assumed.

Minimum value is 1, maximum value is 15.

RECEIVERID=
Specifies the NetView receiver identifier. If omitted, the NETVIPPI is assumed.

DEFAULTRC=
Specifies the return code that the AOI exit should use when a message is not found in the TYPE=SEARCH entries.

If CANCEL is specified, the AOI exit routine will stop IMS from sending additional message segments.

Specifying CANCEL provides the best performance.

If IGNORE is specified, the IMS Automation exit will receive additional segments, but will not act on the messages.

IGNORE is provided for situations where other AOI exits need to see the messages even when the IMS Automation exit does not need to process the message.

If omitted, DEFAULTRC=CANCEL is assumed.

TRANID=
Specifies the IMS transaction identifier to queue messages from the automated operator exit.

If omitted, EVITPPI1 is the default.

SENDERID=
Specifies the IMS sender identifier.

Attention: The sender ID must be the IMS applid.

If omitted, the applid is assumed.

PPIPROC=
Specifies the name of the procedure to be started as a message driven BMP.

This task will handle the transaction identified in TRANID=*keyword*.

If omitted, PPI*ims_id* is the default used.

EXITNAME=
Specifies the name of the user-defined load module that the IMS Automation AOI exit should call.

The exit must follow the IMS DFSAOUE0 coding requirements.

EXITNAME is not required.

If omitted, EXITNAME will default to EVIAOUE0.

TYPE=SEARCH
Describes the search criteria and action to be performed on a message routed to the master terminal operator. The valid parameters for TYPE=SEARCH are:

MSGID=
Specifies the message identifier to search for.

SUP=
Specifies whether to suppress the message if it is matched (YES), or not to suppress the message if it is matched (NO). If omitted, a default value is assumed (NO).

AUTO=
Specifies whether to automate the message by sending it to NetView (YES) or not to automate the message (NO). If omitted, a default value is assumed (YES).

TEXT=
Specifies the text string to look for in the incoming message.

WTO=
Specifies whether to write the message to the system console (YES) or not to write the message (NO). If omitted, a default value is assumed, (NO).

TYPE=ENTRY
Relates function name to an IMS command name.

The TYPE default is ENTRY.

FUNCTION=
Specifies the name of a function

A request for this function causes the related IMS transaction to be executed.

TRANSID=
Specifies the IMS transaction name related to the function specification.

TYPE=FINAL
Generates the initialization table. The TYPE=FINAL keyword is required and must be the last TYPE= keyword in the EVISPINM member.

Step 2: Extend EVINTASK, If Required

EVINTASK is the IMS Automation PPI initialization member on NetView. EVINTASK is used on the NetView side of the PPI to:

1. Set the program-to-program interface buffer queue limit. This is the number of outstanding buffers that can be stored in the receiver buffer queue.
2. Define receiver programs and automation operator tasks to be used for specific functions.
3. Define the PPI receiver ID for the EVISNPPI NetView subtask program.

Figure 18 shows an example of the information contained in EVINTASK :

```
BUFFQL=20
*
RECEIVERID=NETVIPPI
*
SERVER=SEND,MESSAGE,AUTIPPI,EVISNMSG
SERVER=RESPONSE,IMSCMD,AUTIPPI,EVISNRSP
SERVER=RESPONSE,NACK,AUTIPPI,EVISNACK
SERVER=REQUEST,NACK,AUTIPPI,EVISNACK
*
```

Figure 18. EVINTASK PPI Initialization Member

At least one valid SERVER= entry must be specified in EVINTASK. There must also be a SERVER= entry for each function that uses the IMS Automation PPI.

Usually, you will not change the EVINTASK PPI initialization member. However, you must change EVINTASK under these circumstances:

- If you must use a naming convention other than the default names for the automated operator tasks. (The default operator ID IMS Automation uses is AUTIPPI). It is recommended that you uses the default names if possible.
- If your MVS system has both a test NetView and a production NetView running on the same host. Each NetView requires its own receiver ID. The receiver IDs are defined in both EVINTASK and in the corresponding EVISPINM or EVII*ims_id*.

Attention: Take care if you must change the operator IDs for the automated operator tasks or the receiver IDs. Errors when changing these can stop IMS Automation from working.

The keywords in the EVINTASK member are as follows:

BUFFQL=
: PPI buffer queue limit for EVINTASK. Unless you have received a buffer error, do not change this value. To change the limit, specify a 2- or 3-digit number between 10 and 999. If no BUFFQL parameter is coded, the default used is 15.

  For further information, refer to the description of buffer queue limit in the *NetView Programming: Program-to-Program Interface* documentation.

SERVER=

Do not change the SERVER= parameters (with the possible exception of the operator ID, AUTIPPI). The SERVER= parameters define:

1. The first parameter defines whether this function is a REQUEST, RESPONSE, or SEND. A REQUEST is used to identify a receiver program to be invoked if NetView gets a CONVERSE or SEND from IMS. A RESPONSE is used to identify a sender program to be invoked if IMS sends a RESPONSE. The function name cannot begin with the prefix EVI. Do not change this parameter.
2. The second parameter defines the operator ID under which the program runs. The default name is AUTIPPI. Unless the naming convention at your site will not allow it, use the default name. To change the name of the default autotask, use the SA OS/390 customization dialogs to change the AUTOOPS entry (or edit the AUTOOPS control file entry). Change the AUTIPPI to the name required at your site in the following AUTOOPS entry:

   ```
   AUTOOPS    IMSPPI,ID=(AUTIPPI)
   ```

   Change AUTIPPI in the SERVER= entry to match the operator ID you define in the AUTOOPS control file entry.

   Note: If you change the operator ID for any automation operator, make sure that the new ID is correctly defined in DSIOPF member. Use the AUTPPI operator ID as an example.
3. The command list or command processor used for this function, such as EVISNRSP (the common response handler). Do not change this parameter.

RECEIVERID=

The PPI receiver ID for NetView uses to sign on to the PPI. If omitted, NETVIPPI is assumed. Do not change the receiver ID unless necessary. If you run both a test NetView and a production NetView within the same MVS system, you must give each NetView its own receiver ID and change this field to match. In addition, you must change the RECEIVERID= field in the corresponding EVISPINM or EVII*ims_id* member.

Figure 19 shows an MVS system with both a test and production NetView (if you run both in the same MVS system):

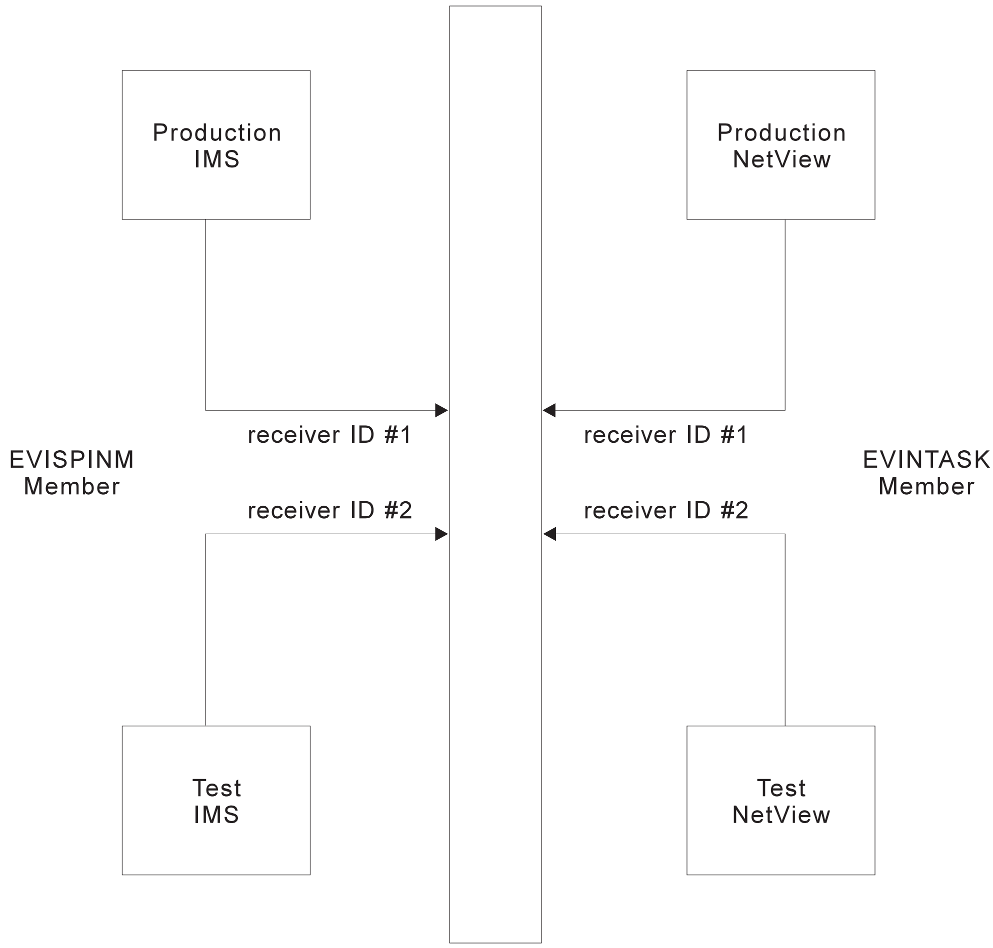

Figure 19. MVS System with Test and Production NetView. Define different receiver IDs for the test and production NetViews

Chapter 11. Testing IMS Automation

This section describes tasks that should be performed to verify that the Automation and Productivity Enhancements have been installed and customized properly. This Installation Verification Procedure (IVP) can also be run after the application of maintenance to the NetView Program Product or the Automation and Productivity Enhancements.

During IVP testing, the following manuals should be used:

- *AOC/MVS IMS Automation Feature Operator's Guide*.
- *AOC/MVS IMS Automation Feature Programmer's Reference and Installation Guide*.

After completing customization of a non-XRF subsystem, perform the test scenarios in Table 21 on page 226 and Table 23 on page 228.

Upon completion of the IVP, review the test results. Document any unusual test results. If possible, provide explanations for these results. If problems have been identified with the test scenario, review the installation steps before contacting the IBM Support Center. In particular, ensure that the control file and NetView automation table have been properly customized.

Startup Test Scenario

Table 21. Startup Test Scenario

Action	Result
Log on to NetView, then enter NCCF. When you are logged on to NCCF, enter IMS.	The IMS Automation: Main Menu is displayed.
Enter `?` in the **Subsystem Name** field and press the Enter key.	The Subsystem List panel displays all of the defined IMS subsystems.
Select a non-XRF subsystem. Type an `S` before a listed non-XRF subsystem and press the Enter key.	The Main Operator Menu panel is displayed. The **Subsystem Name** field is filled in with the subsystem name that you have just selected.
Type `1` in the **Select an option** field and press the Enter key.	The Inquire Subsystem Components panel displays.
Type `2` in the **Select an option** field to display defined regions information and press the Enter key.	The Subsystem/Defined Regions Display panel displays. The current information for the subsystem and its dependent regions is displayed. The status of the subsystems and its dependent regions will be DOWN because they have not been started yet.
Press PF4 to return. Select IMS again.	The Main Operator Menu panel displays. The **Subsystem Name** field still contains the subsystem name that you have selected.
Select option 2 to initiate startup.	The Start Functions panel displays.
Type `1` in the **Select a start type** field., and press the `Enter` key.	This action initiates a default startup, as defined in the control file. The Start Confirmation panel displays, showing the subsystems to be started.
Type `1` and press the Enter key to continue startup.	The Start Confirmation panel will display, showing the subsystems that started successfully.
Press Function Key 4 to return to the IMS Automation main menu.	The IMS Automation: Main Menu panel displays.
Type `1` in the **Select an option** field and press the Enter key.	The Inquire Subsystem Components panel displays.
Enter `2` in the **Select an option** field to display defined regions information.	The Subsystem/Defined Regions Display panel displays. Note that the status of the subsystem and affected dependent regions has changed.
Press Function Key 5 to refresh the panel. Continue to refresh the panel until the IMS subsystem's status is UP.	When the status has been changed to UP for the subsystem, the subsystem is up and functional.
Enter `BR NETLOGA` on the command line to display the active NetView log.	The NetView log displays. Review the messages generated during startup.
You have completed the startup test. Press Function Key 2 to end this IMS Automation session.	You return to NetView.

Message Region Test Scenario

Table 22. Message Region Test Scenario	
Action	**Result**
Log on to NetView, then enter NCCF. When you are logged on to NCCF, enter IMS.	The IMS Automation: Main Menu is displayed.
Enter ? in the **Subsystem Name** field and press the Enter key.	The Subsystem List panel displays all of the defined IMS subsystems.
Select a subsystem. Type an S before a listed non-XRF subsystem and press the Enter key.	The Main Operator Menu panel is displayed. The **Subsystem Name** field is filled in with the subsystem name that you have just selected.
Type 1 in the **Select an option** field and press the Enter key.	The Inquire Subsystem Components panel displays.
Type =6.3 on the command line and press Enter to fastpath to the Message Region Operator panel.	The IMS Automation Dependent Region panel displays. If the control region has a status other than UP, return and start the subsystem. If the status is UP, continue with the test procedure.
To select a Message Region, enter either an I or T beside the region name. Use an I if the message region is not UP, and use a T if the message region is UP.	IMS Automation will process your request to either initiate (I) or terminate (T) a message region. Operator messages display beside the message region name indicating successful completion of request.
Press PF5 to refresh the Operator Panel.	The updated status of your message regions displays on the Operator Panel.
You have completed the message regions test. Press PF2 to end this IMS Automation session. You have tested the program-to-program interface (PPI) of IMS Automation, which sends the requests for initiation or termination of message regions to the IMS control region.	You return to NetView.

Shutdown Test Scenario

Table 23. Shutdown Test Scenario

Action	Result
If you are not already logged on, log on to NetView. Then enter `NCCF`. Once you are logged on to NCCF, enter `IMS`.	The Main Operator Menu panel is displays.
Type `?` in the **Subsystem Name** field and press the Enter key.	The Subsystem Selection panel displays.
Type `1` in the **Select an option** field and press the Enter key.	The Subsystem List panel will display all of the IMS subsystems defined.
Select the subsystem you wish to shut down by typing an **S** beside that subsystem name. Press the Enter key.	The main menu displays with the subsystem name you have just selected in the **Subsystem, Group, or Domain** field.
Type `3` in the **Select an option** field.	The Operator Shutdown panel displays.
Type `1` to initiate a normal shutdown, and press the Enter key.	The Shutdown Confirmation panel displays to verify that the subsystem displayed is the correct subsystem to be shut down. The Restart and Broadcast fields have default values of **N** preselected.
Type `1` in the **Select an option** field, and press the Enter key to proceed with the shutdown.	IMS Automation initiates the shutdown process for the selected subsystem. The Shutdown Notification panel displays, showing that the shutdown command was successfully initiated.
Press Function Key 4 to return to the IMS Automation Main Menu.	The IMS Automation: Main Menu panel displays.
On the command line, enter `=1.2` and press the Enter key.	The Subsystem/Defined Regions Display panel displays. Note that the status of the subsystem and affected dependent regions has changed.
Press Function Key 5 to refresh the panel. Continue to refresh the panel until the IMS subsystem's status is AUTODOWN.	When the status has been changed to AUTODOWN, the subsystem is DOWN and not functional.
You have completed the shutdown test. Press Function Key 2 to end this IMS Automation session.	You return to NetView.

Part 4. Appendixes

Appendix A. MVS Sample JCL

This appendix lists the JCL that is used to customize IMS Automation.

Table 24. Sample JCL

Name	Description	See Page
EVISJ010	Copy AEVISAMP members to NetView DSIPARM data set.	232
EVISJ020	Assemble and link-edit EVISPINM tables.	234

Copy IMS Automation Feature AEVISAMP Members to NetView DSIPARM Data Set

EVISJ010

```
//EVISJ010 JOB 'ACCOUNTING INFORMATION','COPY DSIPARM MEMBERS',
//  CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1)          ,TYPRUN=HOLD
//*******************************************************************
//* (C) COPYRIGHT IBM CORP. 1990, 1995 - ALL RIGHTS RESERVED         *
//*******************************************************************
//*                                                                  *
//*      LICENSED MATERIALS - PROPERTY OF IBM                        *
//*      RESTRICTED MATERIALS OF IBM                                 *
//*      5685-151                                                    *
//*      (C) COPYRIGHT IBM CORP. 1990, 1995                          *
//*                                                                  *
//*      US GOVERNMENT USERS RESTRICTED RIGHTS -                     *
//*      USE, DUPLICATION OR DISCLOSURE RESTRICTED BY                *
//*      GSA ADP SCHEDULE CONTRACT WITH IBM CORP.                    *
//*                                                                  *
//*  APAR#     DATE                                                  *
//* -------  --------                                                *
//*******************************************************************
//*                                                                  *
//*  DESCRIPTION: JCL TO COPY IMSAO SAMPLIB MEMBERS TO THE NETVIEW    *
//*               DSIPARM DATASET                                    *
//*                                                                  *
//*******************************************************************
//* NOTE 1) CHANGE Q1 TO MATCH THAT OF THE ACTUAL HIGH LEVEL QUALIFIER
//*         USED TO ALLOCATE THE IMSAO DATA SETS
//*      2) UPDATE THE DSIPARM DD STATEMENT TO POINT TO THE DSIPARM
//*         DATASET WHERE IMSAO WILL BE TESTED
//*      3) NORMAL JOB CONDITION CODE IS ZERO(0).
//*
//*******************************************************************
//COPYPARM  PROC Q1='AOCIMS.V1R4M0',            HIGH LEVEL QUALIFIER
//          DSIPARM='CNM.SA01.DSIPARM'    DSIPARM DATA SET NAME
//LOADCOPY  EXEC PGM=IEBCOPY,REGION=512K
//SYSPRINT  DD SYSOUT=*
//*****************************************************************
//*    COPY SAMPLIB MEMBERS TO THE DSIPARM DATASET                *
//*****************************************************************
//SEVISAMP  DD DSN=&Q1..SEVISAMP,DISP=SHR
//DSIPARM   DD DSN=&DSIPARM,DISP=SHR
//SYSUT2    DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT3    DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//  PEND
//COPY     EXEC  COPYPARM
//SYSIN    DD  *
  COPY     INDD=SEVISAMP,OUTDD=DSIPARM
  SELECT   MEMBER=(EVIS2CFG,EVIS2STA,EVIS2IMS,EVIS2CRT,EVIS2LCL)
  SELECT   MEMBER=EVIS2DBC
  SELECT   MEMBER=EVISECUR
  SELECT   MEMBER=EVINTASK
  SELECT   MEMBER=EVID0001
  SELECT   MEMBER=(EVICMD,EVIOPF,EVIDMN)
```

```
  SELECT   MEMBER=(EVIDARCA,EVIDARCB,EVIDARC1,EVIDARC2)
  SELECT   MEMBER=(EVIDMSCA,EVIDMSCB,EVIDMSC1,EVIDMSC2)
  SELECT   MEMBER=(EVIDMSGA,EVIDMSGB,EVIDMSG1,EVIDMSG2)
  SELECT   MEMBER=(EVIDOLDA,EVIDOLDB,EVIDOLD1,EVIDOLD2)
  SELECT   MEMBER=(EVIDRECA,EVIDRECB,EVIDREC1,EVIDREC2)
  SELECT   MEMBER=(EVIDTIMA,EVIDTIMB,EVIDTIM1,EVIDTIM2)
  SELECT   MEMBER=(EVIDTRNA,EVIDTRNB,EVIDTRN1,EVIDTRN2)
  SELECT   MEMBER=(EVISS002,EVISS003,EVISS005)
  SELECT   MEMBER=((EVIMSG00,IMSMSG00),EVIMCON0,EVIMEVI0)
  SELECT   MEMBER=((EVIMSG01,IMSMSG01))
  SELECT   MEMBER=(EVIMAVM1,EVIMCON1,EVIMDFS1,EVIMDSP1)
  SELECT   MEMBER=(EVIMDXR1,EVIMEVI1,EVIMIOS1)
//*
```

Assemble and Link-Edit EVISPINM Tables

EVISJ020

```
//*******************************************************************
//*                                                                 *
//*      LICENSED MATERIALS - PROPERTY OF IBM                       *
//*      5685-151                                                   *
//*      (C) COPYRIGHT IBM CORP. 1990, 1996                         *
//*                                                                 *
//* APAR#     DATE                                                  *
//*-------  --------                                                *
//*$L1=ARM  05/31/96    LK CHANGE LIB2=IMS.USERLNK TO IMS.RESLIB    *
//*$01=PN50936 01/03/94 MP INSTALLATION JOB CONTROL REFERENCES      *
//*                        NON-SMPE LIBRARIES                       *
//*                                                                 *
//*******************************************************************
//*  DESCRIPTION: THIS JOB ASSEMBLES AND LINK-EDITS USER MAINTAINED *
//*               EVISPINM TABLES.                                  *
//*******************************************************************
//* NOTE 1) UPDATE DATASET NAMES, VOL=SER, AND UNIT FIELDS.         *
//*      2) IF NECESSARY, UPDATE THE SOURCE AND LOADLIB DATASET NAMES *
//*      3) VERIFY THAT BLKSIZE OF FIRST DATA SET OF SYSLIB CONCAT  *
//*         IN ASM STEP IS LARGEST.                                 *
//*      4) UPDATE SYSLMOD DATA SET TO POINT TO ACTUAL USER LINK LIB *
//*      5) UPDATE NAME1 ANE NAME2 TO POINT TO THE IMS PROCESSING   *
//*         TABLE.  EX. A. EVIIXXXX FOR THE SPECIFIC IMS PROCESSING *
//*                        WHERE XXXX IS THE IMSID OF THE IMS       *
//*                                   THAT OWNS THIS TABLE          *
//*                     B. EVISPINM FOR THE GENERIC IMS PROCESSING  *
//*******************************************************************
//IMSLNK  PROC Q1='AOCIMS.V1R4M0',    * AOC/IMS HIGH LEVIL QUALIFIER
//             LIB1='AOCIMS.V1R4M0.SEVISAMP',  * INPUT SOURCE LIBRARY
//             NAME1=,                * INPUT SOURCE NAME
//             LIB2='IMS.RESLIB',     * LINK-EDIT OUTPUT LIBRARY
//             NAME2=                 * LINK-EDIT OUTPUT NAME
//**************************************************************
//* ASSEMBLY STEP                                              *
//**************************************************************
//ASM     EXEC PGM=IEV90,
//             REGION=3M,
//             PARM='DECK,NOOBJECT,LIST,ALIGN,BATCH'
//SYSLIB    DD DSN=&Q1..SEVISAMP,DISP=SHR
//          DD DSN=SYS1.MACLIB,DISP=SHR
//SYSUT1    DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT2    DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT3    DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSPUNCH  DD DSN=&&LOADSET,
//             UNIT=SYSDA,DISP=(,PASS),
//             SPACE=(400,(100,100,1))
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DSN=&LIB1(&NAME1),DISP=SHR
//**************************************************************
//* LKED                                                       *
//**************************************************************
//LKED    EXEC PGM=IEWL,PARM='RENT,MAP,XREF,LIST',
//             COND=(8,LE,ASM),REGION=512K
```

```
//SYSLIB    DD DSN=&Q1..SEVIMOD2,DISP=SHR
//SYSLMOD   DD DSN=&LIB2(&NAME2),DISP=SHR
//SYSUT1    DD DSN=&&SYSUT1,UNIT=SYSDA,
//             SPACE=(1024,(200,50),,CONTIG,ROUND),DCB=BLKSIZE=1024
//SYSPRINT  DD SYSOUT=*
//SYSLIN    DD DSN=&&LOADSET,DISP=(OLD,DELETE)
//   PEND
//EVISPINM EXEC IMSLNK,NAME1=EVISPINM,NAME2=EVISPINM,
//             LIB2='IMS.RESLIB'
```

Appendix B. Automation and Productivity Enhancements - Examples

This appendix lists examples for the automation and productivity enhancements.

Table 25 (Page 1 of 2). Automation and Productivity Enhancements—Examples

Name	**Description**	**See Page**
EVISPINM	IPDS Initialization Table	239
EVINTASK	NetView PPI Initialization Member	243
IMSMSG00 (EVIMSG00)	Message Automation Table (startup) %INCLUDES the following: • EVIMCON0 (Conflict Messages) • EVIMEVI0 (IMS Automation Messages)	245
IMSMSG01 (EVIMSG01) %INCLUDES the following: • EVIMAVM1 (Restart Manager Messages) • EVIMCON1 (Conflict Messages) • EVIMDFS1 (IMS Messages) • EVIMDSP1 (IMS Messages) • EVIMDXR1 (IRLM Messages) • EVIMCQS1 (IMS CQS Messages) • EVIMEVI1 (IMS Automation Messages) • EVIMIOS1 (IOS Messages)	Message automation table (after startup)	248
EVIOPF	Operator Definitions	269
EVICMD %INCLUDES the following: • EVISECUR (Security Definitions)	Command Model Entries	270
EVIDMN	DSIDMN Entries	274
EVISTSM	Sample DSTINIT	275

Table 25 (Page 2 of 2). Automation and Productivity Enhancements—Examples		
Name	**Description**	**See Page**
EVIS2CFG	Configuration Table Entries—Base Automation %INCLUDES the following: • EVIS2STA (State Action Entries) • EVIS2OPS (AUTOOPS Entries) • EVIS2IMS (IMS Control Region Entries) • EVIS2DBC (DB Control Region Entries) • EVIS2XRF (XRF Control Region Entries) • EVIS2SPX (Sysplex Related Entries) • EVIS2FDR (FDR Related Entries) • EVIS2CQS (CQS Related Entries) • EVIS2CRT (SDF Entries) • EVIS2LCL (Resident Entries)	276
EVIPRFAO	Profile for one or more Automated Operators	329

Required NetView Automation Table Entries

EVISPINM

```
****************************************************************** *
* (C) COPYRIGHT IBM CORP. 1990, 1995 - ALL RIGHTS RESERVED          *
****************************************************************** *
*                                                                   *
*        LICENSED MATERIALS - PROPERTY OF IBM                       *
*        RESTRICTED MATERIALS OF IBM                                *
*        5685-151                                                   *
*        (C) COPYRIGHT IBM CORP. 1990, 1995                         *
*                                                                   *
*        US GOVERNMENT USERS RESTRICTED RIGHTS -                    *
*        USE, DUPLICATION OR DISCLOSURE RESTRICTED BY               *
*        GSA ADP SCHEDULE CONTRACT WITH IBM CORP.                   *
*                                                                   *
****************************************************************** *
*   APAR#     DATE                                                  *
*  -------  --------                                                *
*  $04=OW30328,V1R4,11Nov97,APC(AE): Indicate that DFS551I needs to be*
*                                    hard-coded.                    *
*                                                                   *
*  $03=PN49361 11/18/93 MP IMSAO SUPPORT FOR IMS V4R1               *
*                                                                   *
*  $02=PN35803 11/16/93 MP EVISPPII (EVI2PPII) + X'622' AFTER TKO.  *
*                          ADDED COMMENT BEFORE DFS551I, REMOVED    *
*                          DFS994 MESSAGE AND COMMENT FROM TABLE    *
*                          REMOVED IMSID=???? FROM TYPE=INITIAL     *
*                          MACRO, LEFT COMMENT OF IMSID=????        *
*                                                                   *
*  $01=PN36820 04/19/93 MP IMSAO STATE/ACTION TABLE UPDATES AND     *
*                          CORRECTIONS.                             *
*                                                                   *
*  PN26305  11/24/92   RW  IMSAO SAMPLE MEMBER EVISPINM LACKS CRITICAL*
*                          ENTRY FOR IMSID ON TYPE=INITIAL INVOCATION *
*                          OF THE EVIMPINM MACRO                    *
*                                                                   *
*  PN29214  10/23/92   MP  IMSAO AOI EXIT EVISAOIX SETTING RC 12 FOR  *
*                          MESSAGES ON THE AOI                      *
*                                                                   *
*******************************************************************
*                                                                   *
*  FUNCTION:                                                        *
*          TO GENERATE THE IPDS INITIALIZATION TABLE. THIS          *
*          TABLE DEFINES AMONG OTHERS THE RELATIONSHIP              *
*          BETWEEN FUNCTION NAMES AND IMS TRANSACTION               *
*          NAMES.                                                   *
*                                                                   *
*  ENTRY POINTS:                                                    *
*          EVILPINM                                                 *
*                                                                   *
*  INVOCATION:                                                      *
*          N/A                                                      *
*                                                                   *
*  INPUT:                                                           *
*          N/A                                                      *
```

```
*                                                                      *
*  PROCESS:                                                            *
*          N/A                                                         *
*                                                                      *
*  OUTPUT:                                                             *
*          NONE, THERE IS NO EXECUTABLE CODE.                          *
*                                                                      *
*  ERRORS DETECTED:                                                    *
*          NONE                                                        *
*                                                                      *
*  EXITS                                                               *
*          N/A                                                         *
*                                                                      *
*  REGISTER USAGE:                                                     *
*          N/A                                                         *
*  NOTES                                                               *
*    ATTRIBUTES:                                                       *
*            RENT, AMODE(31),RMODE(ANY), AC=0.                         *
*    RESTRICTIONS:                                                     *
*    DEPENDENCIES:                                                     *
*    OTHER:                                                            *
*            EVISPINM IS TO BE LINK-EDITED IN ONE OF THE               *
*            IMS DFHRPL LIBRARIES.                                     *
*                                                                      *
*            GENERATION OF THE IPDS INITIALIZATION TABLE               *
*            REQUIRES EVIMPINM TYPE=DSECT T BE SPECIFIED.              *
*                                                                      *
************************************************************************
         TITLE '*** IAO *** IPDS INITIALIZATION TABLE - GENERATE TABLE' 00590000
****************************************************************
****************************************************************
*                                                              *
*        SPECIFY DEFINITIONS FOR IPDS INITIALIZATION TABLE     *
*                                                              *
****************************************************************
****************************************************************
         SPACE 2
****************************************************************
*                                                              *
*        INITIAL EVIMPINM SPECIFICATION                        *
*                                                              *
*        'INITIAL' MUST BE THE FIRST EVIMPINM TYPE SPECIFIED.  *        00720000
*                                                              *
*        BUFFQL SPECIFIES THE BUFFER QUEUE LIMIT FOR THE       *
*        IMS RECEIVER SIDE OF THE PPI INTERFACE TO NETVIEW.    *
*        A MINIMUM VALUE OF 1 AND A MAXIMUM VALUE OF 15        *
*        CAN BE SPECIFIED. IF THIS KEYWORD IS OMITTED, A       *
*        DEFAULT VALUE OF 3 IS ASSUMED.                        *
*                                                              *
*        RECEIVERID SPECIFIES THE IDENTIFIER OF THE NETVIEW    *
*        RECEIVER. THIS NAME MUST BE THE SAME AS THE NAME      *
*        SPECIFIED IN THE EVISNPPI INITIALIZATION MEMBER.      *
*        IF THIS KEYWORD IS OMITTED, NETVIPPI IS ASSUMED.      *
*                                                              *
****************************************************************
         SPACE 1
         EVIMPINM TYPE=INITIAL,       INITIAL ENTRY                    *
               BUFFQL=15,             BUFFER QUEUE LIMIT               *
```

```
              RECEIVERID=NETVIPPI,   NPDS RECEIVER IDENTIFICATION     *
              DEFAULTRC=CANCEL,      CANCEL (12) OR IGNORE (04)       *
              TRANID=EVITPPI1        IPDS TRANSACTION NAME
*             IMSID=????             OPTIONAL TABLE SUFFIX       @01C
        EJECT
        SPACE 1
****************************************************************
*                                                              *
*       DEFINE MESSAGES PROCESSING INFORMATION                 *
*                                                              *
*       SPECIFY MESSAGESS FOR PROCESSING BY THE AUTOMATED      *
*       OPERATOR EXIT.                                         *
*                                                              *
****************************************************************
        SPACE 1
* DFS551 IS USED TO COMPLETE PPI CONNECTION DURING IMS INITIALIZATION
* This entry is required - AOC/IMS BMP will not initialize      @04A
* until it receives a DFS551I message containing the AOC/IMS    @04A
* BMP region name.                                              @04A
        EVIMPINM TYPE=SEARCH,MSGID=DFS551I,SUP=NO,AUTO=YES
        EVIMPINM TYPE=SEARCH,MSGID=DFS552I,SUP=NO,AUTO=YES
        EVIMPINM TYPE=SEARCH,MSGID=DFS0414I,SUP=NO,AUTO=YES
        EVIMPINM TYPE=SEARCH,MSGID=DFS2161I,SUP=NO,AUTO=YES
        EVIMPINM TYPE=SEARCH,MSGID=DFS2168I,SUP=NO,AUTO=YES
        EVIMPINM TYPE=SEARCH,MSGID=DFS2169I,SUP=NO,AUTO=YES       @01C
        EVIMPINM TYPE=SEARCH,MSGID=DFS554A,SUP=NO,AUTO=YES
        EJECT
        SPACE 1
****************************************************************
*                                                              *
*       DEFINE FUNCTION AND TRANSACTION NAMES                  *
*                                                              *
*       SPECIFY FOR A FUNCTION THE IMS TRANSACTION TO          *
*       BE EXECUTED WHEN THE FUNCTION IS REQUESTED.            *
*       THE FUNCTION NAME CAN BE FROM 1 TO 8 CHARACTERS.       *
*                                                              *
*       TRANSACTION IS NOT USED CURRENTLY FOR THE IMS          *
*         AUTOMATION OPTION -- RESERVED FOR FUTURE USE         *
*                                                              *
****************************************************************
        SPACE 1
        EVIMPINM TYPE=ENTRY,      DEFINE A FUNCTION                   *
              FUNCTION=IMSCMD,    FUNCTION NAME                       *
              TRANSID=IMS         TRANSACTION NAME
        EJECT
        SPACE 1
****************************************************************
*                                                              *
*       'FINAL' MUST BE THE LAST EVIMPINM TYPE SPECIFIED.      *       01880000
*                                                              *
****************************************************************
        SPACE 1
        EVIMPINM TYPE=FINAL       REQUIRED END
        TITLE '*** IAO *** IPDS INITIALIZATION TABLE - DUMMY SECTION'  01930000
        SPACE 1
        EVIMPINM TYPE=DSECT
        EJECT
        SPACE 1
```

```
END EVILPINM
```

EVINTASK

```
*******************************************************************
*     (C) COPYRIGHT INTERNATIONAL BUSINESS MACHINES CORPORATION 1995 *
*                                                                  *
*******************************************************************
*   APAR#     DATE                                                 *
*  -------  --------                                               *
*  PN03634  10/28/91   RR  WHEN ISSUING COMMANDS TO THE BACKUP, THE PP*
*                          IS ATTEMPTED, BUT THE COMMAND IS ACTUALLY  *
*                          SENT TO THE ACTIVE                         *
*                                                                     *
*******************************************************************
* NAME:                                                            *
*         EVENTASK INITIALIZATION MEMBER                           *
*                                                                  *
*  DESCRIPTION:                                                    *
*                                                                  *
*         THIS SAMPLE CONTAINS SOME DEFINITIONS FOR THE            *
*         EVENTASK INITIALIZATION MEMBERS. THE MEMBER              *
*         MUST BE CREATED IN (ONE OF) THE DSIPARM DATA             *
*         SETS.                                                    *
*                                                                  *
*  NOTES:                                                          *
*    DEPENDENCIES:                                                 *
*           NONE.                                                  *
*    OTHER:                                                        *
*           THIS SAMPLE MUST BE CUSTOMIZED ACCORDING TO THE        *
*           ENVIRONMENT OF THE INSTALLATION.                       *
*                                                                  *
*******************************************************************
*                                                                  *
*                                                                  *
*       AN ASTERISK IN COLUMN ONE INDICATES A COMMENT LINE.        *
*                                                                  *
*                                                                  *
*******************************************************************
*                                                                  *
*       PPI BUFFER QUEUE LIMIT FOR EVINTASK                        *
*                                                                  *
*       SPECIFY A 2-DIGIT OR 3-DIGIT NUMERIC VALUE. THE            *
*       MINIMUM VALUE IS 10, THE MAXIMUM VALUE IS 999.             *
*       IF OMITTED, A VALUE OF 15 IS ASSUMED.                      *
*                                                                  *
*******************************************************************
*
BUFFQL=20
*
*******************************************************************
*                                                                  *
*       PPI RECEIVER IDENTIFIER FOR EVISNPPI TASK                  *
*                                                                  *
*       IF OMITTED, NETVIPPI IS ASSUMED.                           *
*                                                                  *
*******************************************************************
*
RECEIVERID=NETVIPPI
*
```

```
**********************************************************************
*                                                                    *
*        FUNCTION NAMES AND SERVER PROGRAMS                          *
*                                                                    *
*        THE FUNCTION NAME MAY NOT START WITH "EVI".                 *
*        AT LEAST ONE VALID SERVER MUST BE SPECIFIED.                *
*                                                                    *
**********************************************************************
*
SERVER=SEND,MESSAGE,AUTIPPI,EVISNMSG
SERVER=RESPONSE,IMSCMD,AUTIPPI,EVISNRSP
SERVER=RESPONSE,NACK,AUTIPPI,EVISNACK
SERVER=REQUEST,NACK,AUTIPPI,EVISNACK
*
**********************************************************************
*                                                                    *
*        END OF EVINTASK INITIALIZATION MEMBER                       *
*                                                                    *
**********************************************************************
```

IMSMSG00 (EVIMSG00)

```
************************************************************************00010000
* (C) COPYRIGHT IBM CORP. 1990, 1995 - ALL RIGHTS RESERVED             *00020000
************************************************************************00030000
*                                                                      *00040000
*        Licensed Materials - Property of IBM                          *00050000
*        Restricted Materials of IBM                                   *00060000
*        5685-151                                                      *00070000
*        (C) Copyright IBM CORP. 1990, 1995                            *00080000
*                                                                      *00090000
*        US Government Users Restricted Rights -                       *00100000
*        Use, duplication or disclosure restricted by                  *00110000
*        GSA ADP Schedule Contract with IBM Corp.                      *00120000
*                                                                      *00130000
************************************************************************00140000
* Description:  EVIMSG00 - NetView Automation table used during        *00150000
*               NetView intialization.                                 *00160000
*                                                                      *00170000
************************************************************************00180000
*  AT NETVIEW INITIALIZATION OR DOMAIN RESTART THERE MAY OCCUR A       *00190000
*  SITUATION WHEN DSIMSG/TBL00 IS STILL LOADED WHEN GATEWAYS ARE       *00200000
*  STARTED. THIS CAUSES MESSAGE TABLE TRAPS FOR SPOC TO BE MISSED      *00210000
*  AND SPOC NOT TO FUNCTION. THE ENTRIES BELOW PLACED IN DSIMSG/TBL00  *00220000
*  WILL ENSURE THIS DOES NOT OCCUR.  NOTE:  SPOC IS SINGLE POINT OF    *00230000
*  CONTROL                                                             *00240000
************************************************************************00250000
*  IMS AUTOMATION MESSAGES                                             *00260000
************************************************************************00270000
*                                                                       00280000
 %INCLUDE EVIMEVI0                                                      00290000
*                                                                       00300000
************************************************************************00310000
*  IMS AUTOMATION CONFLICT MESSAGES                                    *00320000
************************************************************************00330000
*                                                                       00340000
 %INCLUDE EVIMCON0                                                      00350000
*                                                                       00360000
************************************************************************00370000
*  END OF IMS AUTOMATION MESSAGE ENTRIES FOR SINGLE POINT OF CONTROL   *00380000
************************************************************************00390000
```

EVIMCON0

```
***********************************************************************00010000
* (C) COPYRIGHT IBM CORP. 1995 - ALL RIGHTS RESERVED                   *00020000
***********************************************************************00030000
*                                                                      *00040000
*        LICENSED MATERIALS - PROPERTY OF IBM                          *00050000
*        RESTRICTED MATERIALS OF IBM                                   *00060000
*        5685-151                                                      *00070000
*        (C) COPYRIGHT IBM CORP. 1995                                  *00080000
*                                                                      *00090000
*        US GOVERNMENT USERS RESTRICTED RIGHTS -                       *00100000
*        USE, DUPLICATION OR DISCLOSURE RESTRICTED BY                  *00110000
*        GSA ADP SCHEDULE CONTRACT WITH IBM CORP.                      *00120000
*                                                                      *00130000
***********************************************************************00140000
*   APAR#     DATE                                                     *00150000
*  -------  --------                                                   *00160000
***********************************************************************00170000
* DESCRIPTION:  IMS MESSAGE TABLE FOR THE AOF* MESSAGES                *00180000
*               DURING AUTOMATION START UP                             *00190000
***********************************************************************00200000
*                                                                       00210000
IF MSGID = 'AOF660I' & TEXT = MESSAGE & HDRMTYPE ¬= 'E'                 00220000
   THEN EXEC(CMD('EVIEX041 'MESSAGE) ROUTE(ONE *))                      00230000
        DISPLAY(N) BEEP(N) HOLD(N) NETLOG(N) SYSLOG(N) CONTINUE(Y);     00240000
*                                                                       00250000
IF MSGID = 'AOF662I' & TEXT = MESSAGE & HDRMTYPE ¬= 'E'                 00260000
   THEN EXEC(CMD('EVIEX040 'MESSAGE) ROUTE(ONE *))                      00270000
        DISPLAY(N) BEEP(N) HOLD(N) NETLOG(N) SYSLOG(N) CONTINUE(Y);     00280000
*                                                                       00290000
 IF MSGID='AOF664I' & TEXT= MESSAGE & HDRMTYPE¬='E'                     00300000
   THEN EXEC(CMD('EVIEX042 ' MESSAGE) ROUTE(ALL *))                     00310000
        DISPLAY(Y) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(N) CONTINUE(Y);     00320000
```

EVIMEVI0

```
************************************************************************00010000
* (C) COPYRIGHT IBM CORP. 1995 - ALL RIGHTS RESERVED                   *00020000
************************************************************************00030000
*                                                                      *00040000
*        LICENSED MATERIALS - PROPERTY OF IBM                          *00050000
*        RESTRICTED MATERIALS OF IBM                                   *00060000
*        5685-151                                                      *00070000
*        (C) COPYRIGHT IBM CORP. 1995                                  *00080000
*                                                                      *00090000
*        US GOVERNMENT USERS RESTRICTED RIGHTS -                       *00100000
*        USE, DUPLICATION OR DISCLOSURE RESTRICTED BY                  *00110000
*        GSA ADP SCHEDULE CONTRACT WITH IBM CORP.                      *00120000
*                                                                      *00130000
************************************************************************00140000
*   APAR#     DATE                                                     *00150000
*  -------  --------                                                   *00160000
************************************************************************00170000
* DESCRIPTION:  MESSAGE TABLE ENTRY FOR IMS SINGLE POINT OF CONTROL    *00180000
* AND IMS SINGLE POINT OF CONTROL SUPPORT MESSAGES                     *00190000
*                                                                      *00200000
************************************************************************00210000
*                                                                       00220000
* CROSS-SYSTEM MESSAGE ROUTING                                          00230000
 IF MSGID  = 'EVI697'                                                   00240000
   THEN EXEC( CMD('EVISX003 ') ROUTE(ONE *))                            00250000
        DISPLAY(Y) NETLOG(N);                                           00260000
*                                                                       00270000
* SINGLE POINT OF CONTROL                                               00280000
 IF MSGID  = 'EVI895I' & TEXT = MESSAGE                                 00290000
   THEN EXEC( CMD('EVISXDMB 'MESSAGE) ROUTE(ONE *))                     00300000
        DISPLAY(Y) NETLOG(N);                                           00310000
*                                                                       00320000
```

IMSMSG01 (EVIMSG01)

```
************************************************************************00010000
* (C) COPYRIGHT IBM CORP. 1990, 1995 - ALL RIGHTS RESERVED              *00020001
************************************************************************00030000
*                                                                      *00040000
*        Licensed Materials - Property of IBM                          *00050000
*        Restricted Materials of IBM                                   *00060000
*        5685-151                                                      *00070000
*        (C) Copyright IBM CORP. 1990, 1995                            *00080001
*                                                                      *00090000
*        US Government Users Restricted Rights -                       *00100000
*        Use, duplication or disclosure restricted by                  *00110000
*        GSA ADP Schedule Contract with IBM Corp.                      *00120000
*                                                                      *00130000
************************************************************************00140000
*   APAR#     DATE                                                     *00150000
*  -------  --------                                                   *00160000
*  $01=OW35606,V1R4,17Jan99,APC(NS):Cater for CQS processing.          *
************************************************************************00170000
* DESCRIPTION:  MESSAGE TABLE ENTRY FOR IMS SINGLE POINT OF CONTROL    *00180000
* AND IMS SINGLE POINT OF CONTROL SUPPORT MESSAGES                     *00190000
*                                                                      *00200000
************************************************************************00210000
************************************************************************00220000
*  INCLUDE FOR EVI* MESSAGES                                           *00230000
************************************************************************00240000
*                                                                       00250000
%INCLUDE EVIMEVI1                                                       00260000
*                                                                       00270000
************************************************************************00280000
*  INCLUDE FOR AVM* MESSAGES                                           *00290000
************************************************************************00300000
*                                                                       00310000
%INCLUDE EVIMAVM1                                                       00320000
*                                                                       00330000
************************************************************************00340000
*  INCLUDE FOR CQS* MESSAGES  MERGE this Include if CQS enabled   @01A *00350000
************************************************************************00360000
*                                                                       00370000
%INCLUDE EVIMCQS1                                                        0038000
*                                                                       00390000
*                                                                       00390000
************************************************************************00340000
*  INCLUDE FOR DFS* MESSAGES                                           *00350000
************************************************************************00360000
*                                                                       00370000
%INCLUDE EVIMDFS1                                                       00380000
*                                                                       00390000
*                                                                       00390000
************************************************************************00400000
*  INCLUDE FOR DSP* MESSAGES                                           *00410000
************************************************************************00420000
*                                                                       00430000
%INCLUDE EVIMDSP1                                                       00440000
*                                                                       00450000
************************************************************************00460000
*  INCLUDE FOR IOS* MESSAGES                                           *00470000
```

```
************************************************************************00480000
*                                                                       00490000
%INCLUDE EVIMIOS1                                                       00500000
*                                                                       00510000
************************************************************************00520000
*  INCLUDE FOR DXR* MESSAGES                                           *00530000
************************************************************************00540000
*                                                                       00550000
%INCLUDE EVIMDXR1                                                       00560000
*                                                                       00570000
************************************************************************00580000
*  INCLUDE FOR CONFLICT MESSAGES                                       *00590000
************************************************************************00600000
*                                                                       00610000
%INCLUDE EVIMCON1                                                       00620000
*                                                                       00630000
************************************************************************00640000
*  SAMPLE ENTRY FOR SENDING MESSAGES TO THE CRITICAL MESSAGE MANAGER   *00650000
************************************************************************00660000
*                                                                       00670000
* IF MSGID = 'ANYMSG' & TEXT= MESSAGE & DOMAINID = %AOFDOM% THEN        00680000
*   EXEC(CMD('IMSFWM ' MESSAGE ) ROUTE(ONE *) ) ;                       00690000
*                                                                       00700000
************************************************************************00710000
*  END OF IMS AUTOMATION MESSAGE ENTRIES                               *00720000
************************************************************************00730000
```

EVIMAVM1

```
*********************************************************************00010000
* (C) COPYRIGHT IBM CORP. 1995 - ALL RIGHTS RESERVED                *00020001
*********************************************************************00030000
*                                                                   *00040000
*       Licensed materials - Property of IBM                        *00050000
*       Restricted materials of IBM                                 *00060000
*       5685-151                                                    *00070000
*       (C) Copyright IBM Corp. 1995                                *00080001
*                                                                   *00090000
*       US government users restricted rights -                     *00100000
*       use, duplication or disclosure restricted by                *00110000
*       GSA ADP schedule contract with IBM corp.                    *00120000
*                                                                   *00130000
*********************************************************************00140000
*   APAR#     DATE                                                  *00150000
*  -------  --------                                                *00160000
* $01=OW35606,V1R4,17Nov98,APC(NS):Invoke new module for FDR processing*
*********************************************************************00170000
* Description:  IMS message table for the AVM* messages             *00180000
*********************************************************************00190000
 IF MSGID = 'AVM' .                                                  00200000
   THEN                                                              00210000
     BEGIN;                                                          00220000
*                                                                    00230000
 IF MSGID = 'AVM005A'                                                00240000
    & TEXT = MESSAGE                                                 00250000
    & DOMAINID = %AOFDOM%                                            00260000
    THEN EXEC(CMD('EVIE0007 ' MESSAGE) ROUTE(ALL *))                 00270000
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);        00280000
*
* This entry replaces the msg AVM006E trap which was used for XRF  @01A 00290000
* processing executing clist EVIE0009. For FDR support a new       @01A 00290000
* clist EVIAVM06 has been introduced to handle both XRF & FDR      @01A 00290000
* processing.                                                      @01A 00290000
*
 IF MSGID = 'AVM006E'                                                00300000
    & TEXT = MESSAGE                                                 00310000
    & DOMAINID = %AOFDOM%                                            00320000
    THEN EXEC(CMD('EVIAVM06 ' MESSAGE) ROUTE(ALL *))                 00330000
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);        00340000
*                                                                    00350000
 IF MSGID = 'AVM010E'                                                00360000
    & DOMAINID = %AOFDOM%                                            00370000
    THEN EXEC(CMD('EVIET00E') ROUTE(ALL *))                          00380000
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);        00390000
*                                                                    00400000
  ALWAYS;                                                            00410000
 END;                                                                00420000
```

EVIMCON1

```
*********************************************************************00010002
*********************************************************************00020002
*                                                                   *00030002
*       LICENSED MATERIALS - PROPERTY OF IBM                        *00040002
*       5685-151                                                    *00050002
*       (C) COPYRIGHT IBM CORP. 1994, 1996                          *00060002
*                                                                   *00070002
*********************************************************************00080002
*     APAR#      Date   By Description                              *00090002
*  ----------- -------- -- ---------------------------------------- *00100002
*  $03=OW35553,V1R4,26OCT97,APC(NS): Hold data and Documentation       *
*                                    missing from apar OW27199         *
*                                                                      *
*  $02=OW20827 06/05/96 CG PTF UW90273 DID NOT INCLUDE ++HOLD ACTION   *
*                          STATEMENTS INSTALLER MAY MISS NECESSARY     *
*                          INSTALLATION INFORMATION                    *
*                                                                      *
*  $L1=OW19977 05/31/96 LK AOC IMS-AUTOMATION FEATURE SUPPORT FOR IMS  *
*                          5.1 ARM RECOVERY                            *
*                                                                      *
*  $01=OW07849 09/15/94 JS AOC IMS INITIALIZATION PROBLEMS AFTER AOC   *00120002
*                          APAR OW03453                                *00130002
*                                                                      *00140002
*********************************************************************00150002
* -----------------------------------------------------------------*  00160000
* IMS STARTUP CONTROL                                               *  00170000
*-------------------------------------------------------------------*  00180000
*                                                                      00190000
* MVS DISPLAY ACTIVE                                                   00200000
* WARNING ----POSSIBLE CONFLICT WITH EXISTING MESSAGE TABLE STMTS      00210000
*                                                               @01A   00220002
 IF (MSGID='IEE105I'   MSGID='IEE115I') & TEXT=MESSAGE                 00230000
   THEN EXEC(CMD('EVIEIMON 'MESSAGE)  ROUTE(ALL *))                    00240000
              DISPLAY(Y) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y)           00250000
              CONTINUE(Y);                                             00260000
*                                                                      00270000
* SUBSYSTEM IN CTLDOWN STATUS NOT STARTED                              00280000
* WARNING ----POSSIBLE CONFLICT WITH EXISTING MESSAGE TABLE STMTS      00290000
*                                                               @01A   00300002
 IF MSGID='AOF313I' & TEXT = . 'CTLDOWN' . & TEXT=MESSAGE              00310000
   THEN EXEC(CMD('EVIEIMON 'MESSAGE)  ROUTE(ALL *))                    00320000
              DISPLAY(Y) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y)           00330000
              CONTINUE(Y);                                             00340000
*                                                                      00350000
* WARNING ---- POSSIBLE CONFLICT WITH EXISTING MESSAGE TABLE STMTS     00360000
 IF MSGID = 'IEF450I'                                                  00370000
    & TEXT¬= . ' ABEND=S122 ' . & TEXT¬= . ' ABEND=S222 ' .            00380000
    & TEXT¬= . ' ABEND=S522 ' . & TEXT¬= . ' ABEND=S622 ' .            00390000
    & TEXT = MESSAGE & TOKEN(2) = SVJOB                                00400000
    & (TOKEN(2) = 'IMS'. |                                             00410002
       TOKEN(2) = 'IMS'.)                                              00420002
    & DOMAINID = %AOFDOM%                                              00430000
    THEN EXEC(CMD(' EVISROUT JOBNAME='SVJOB ' EVIER002 ' SVJOB ' '     00440000
               MESSAGE)    ROUTE(ALL *))                               00450000
```

```
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);          00460000
*                                                                      00470000
* WARNING ---- POSSIBLE CONFLICT WITH EXISTING MESSAGE TABLE STMTS @03A 00470000
 IF MSGID = 'IEF404I'                                                  00470000
    & TOKEN(2) = SVJOB                                                 00470000
    & (TOKEN(2) = 'IMS'.                                               00470000
    |  TOKEN(2) = 'IMS'.)                                              00470000
    & DOMAINID = %AOFDOM%                                              00470000
    THEN EXEC(CMD(' EVISROUT JOBNAME='SVJOB                            00470000
         ' TERMMSG FINAL=YES,JOBNAME=' SVJOB)                          00470000
         ROUTE( ALL *))                                                00470000
         DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);               00470000
*                                                                      00480000
****IMS****************************************************************00480000
* MESSAGE TABLE ENTRY FOR IMS SINGLE POINT OF CONTROL                  00490000
***********************************************************************00500000
*                                                                      00510000
IF MSGID = 'AOF660I' & TEXT = MESSAGE & HDRMTYPE ¬= 'E'                00520000
   THEN EXEC(CMD('EVIEX041 'MESSAGE) ROUTE(ONE *))                     00530000
        DISPLAY(N) BEEP(N) HOLD(N) NETLOG(N) SYSLOG(N) CONTINUE(Y);    00540000
*                                                                      00550000
IF MSGID = 'AOF662I' & TEXT = MESSAGE & HDRMTYPE ¬= 'E'                00560000
   THEN EXEC(CMD('EVIEX040 'MESSAGE) ROUTE(ONE *))                     00570000
        DISPLAY(N) BEEP(N) HOLD(N) NETLOG(N) SYSLOG(N) CONTINUE(Y);    00580000
*                                                                      00590000
 IF MSGID='AOF664I' & TEXT= MESSAGE & HDRMTYPE¬='E'                    00600000
   THEN EXEC(CMD('EVIEX042 ' MESSAGE) ROUTE(ALL *))                    00610000
        DISPLAY(Y) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(N) CONTINUE(Y);    00620000
*                                                                      00630000
*                                                                      00640002
*-------------------------------------------------------------------*  00650002
* ARM STARTED JOB                                               @L1A*  00660002
*-------------------------------------------------------------------*  00670002
*                                                                      00680002
 IF MSGID='IXC812I' & TEXT = MESSAGE                                   00690002
   THEN EXEC(CMD('EVIEARMM ' MESSAGE)                                  00700002
                ROUTE(ONE *))                                          00710002
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);          00720002
*                                                                      00730002
*-------------------------------------------------------------------*  00740002
* ARM UNABLE TO START JOB                                       @L1A*  00750002
*-------------------------------------------------------------------*  00760002
*                                                                      00770002
 IF MSGID='IXC804I' & TEXT = MESSAGE                                   00780002
   THEN EXEC(CMD('EVIEARMM ' MESSAGE)                                  00790002
                ROUTE(ONE *))                                          00800002
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);          00810002
```

EVIMDFS1

```
***********************************************************************00000100
* (C) COPYRIGHT IBM CORP. 1996 - ALL RIGHTS RESERVED                  *00000200
***********************************************************************00000300
*                                                                     *00000400
*        Licensed materials - Property of IBM                         *00000500
*        Restricted materials of IBM                                  *00000600
*        5685-151                                                     *00000700
*        (C) Copyright IBM Corp. 1996                                 *00000800
*                                                                     *00000900
*        US government users restricted rights -                      *00001000
*        use, duplication or disclosure restricted by                 *00001100
*        GSA ADP schedule contract with IBM corp.                     *00001200
*                                                                     *00001300
***********************************************************************00001400
*     APAR#      Date   By Description                                *00001500
*  ----------- -------- -- -------------------------------------------*00001600
*                                                                     *
*  $20=OW38079,V1R4,16Mar99,APC(IG):  Add HDRMTYPE filter to          *
*                                     msg DFS058I                     *
*  $19=OW35606,V1R4,11Jan99,APC(NS) : Cater for SAfOS390 V1R3 and IMS  *00020930
*                                     V6 CQS & FDR options            *
*  $18=OW35425,V1R4,02Oct98,APC(GJD): Allow for type 4 in msgDFS554A   *00020930
*                                                                     *
*  $17=OW33665,V1R4,23Jun98,APC(GJD): Update route for msgDFS810A and  *00020930
*                                     change EHKESORP to OUTREP.      *
*                                                                     *
*  $16=OW32145,V1R4,09Mar98,APC(RW): Trap unknown WTOR's                00020930
*  $15=OW25199,V1R4,04Nov97,APC(MJ): OW22099 is a PE. Add back        *
*                                    regressed APARs OW20827, OW19977, *
*                                    and OW16002.                     *
*                                                                     *
*  $14=OW25966,V1R4,01Apr97,APC(AE): Check for 'ABEND' in DFS629I trap *
*                                                                     *
*  $13=OW22405 19961210 WDH MSGDFS554A NOT HANDLED CORRECT IN EVIEY00S *
*                                                                     *
*  $12=OW22099 11/08/96 WDH AOC/MVS IMS, AFTER IMS RESTART, /NRE REPLY *
*                          TO MSGDFS810A HALTS STARTUP WITH MSGDFS3131A*
*                          AND MSGDFS3626I.                           *
*                                                                     *
*  $11=OW21076 09/12/96 HR Remove check for CTL TCB for message DFS629I*
*                                                                     *
*  $10=OW22218 09/02/96 HR Change EHKESGUP to ACTIVMSG and send to     *
*                          GSSOPER                                    *
*                                                                     *
*  $09=OW20827 06/05/96 CG PTF UW90273 DID NOT INCLUDE ++HOLD ACTION   *
*                          STATEMENTS INSTALLER MAY MISS NECESSARY    *
*                          INSTALLATION INFORMATION                   *
*                                                                     *
*  $L1=OW19977 05/31/96 LK AOC IMS-AUTOMATION FEATURE SUPPORT FOR IMS  *
*                          5.1 ARM RECOVERY                           *
*                                                                     *
*  $08=OW19688 04/10/96 CG SAMPLE EVIMDXR1 CONTAINS BEGIN END BLOCK FOR*
*                          DXR BUT WITHIN THE BLOCK THERE ARE COMPARES *
*                          FOR MESSAGE IDS BEGINNING WITH DFS.        *
*                                                                     *
*  $07=OW18749 02/23/96 CG DFS994I IMS (DBCTL) SHUTDOWN COMPLETED      *
```

```
*                          MESSAGE DOESN'T TRIGGER EVIET006 FROM MAT.  *
*                                                                      *
*  $06=OW18339 02/05/96 CG MAT ENTRY IN SEVISAMP MEMBER EVIMDFS1 DOES  *00001700
*                          NOT TRAP DFS3626I WITH HDRMTYPE '>'.        *00001800
*                          MSGDFS3626I MSGDFS166 IMS STARTUP HANG /    *00001900
*                          HUNG                                        *00002000
*                                                                      *00002100
*  $05=OW17788 01/04/96 CG RECEIVED MSGEVI778A DURING RESTART OF       *00002200
*                          FAILED ACTIVE/NEW BACKUP                    *00002300
*                                                                      *00002400
*  $04=OW17107 11/29/95 CG EVIET008 NOT TRAPPING MSGDFS2717 NOR        *00002500
*                          MSGDFS058I STOP COMMAND IN PROGRESS         *00002600
*                                                                      *00002700
*  $03=OW16002 10/05/95 CG Level-set R4.                               *00002800
*                                                                      *00002900
*  $02=OW06699 08/29/95 JS TCO FAILS IN A SINGLE POINT OF CONTROL ENV  *00003000
*                          (SPO).                                      *00003100
*                                                                      *00003200
*  $01=OW11717 03/07/95 JS AOC IMS MESSAGE TABLE DOES NOT ROUTE WTOR   *00003300
*                          HANDLING TO THE CORRECT AUTOTASK AOFOPWTORS *00003400
*                                                                      *00003500
***********************************************************************00003600
* Description:  IMS message table for the DFS* messages                *00003700
***********************************************************************00003800
 IF MSGID = 'DFS' .                                                     00003900
   THEN                                                                 00004000
     BEGIN;                                                             00004100
*                                                                       00004200
* START OF IMS FEATURE MESSAGE TABLE ENTRIES *                          00004300
*                                                                       00004400
IF (MSGID  = 'DFS000I' )                                                00004500
   & DOMAINID = %AOFDOM%                                                00004600
   THEN  EXEC(CMD('EVIEXPIP') ROUTE(ALL AUTIMS)) CONTINUE(Y);           00004700
*                                                                       00004800
IF (MSGID='DFS989I' .    )                                              00004900
 & TEXT= . '(CRC='CRCC')' . '-' SUBID .                                 00005000
   & DOMAINID = %AOFDOM%                                                00005100
    THEN EXEC(CMD('EVIEI200 'CRCC' 'SUBID) ROUTE(ALL *))                00005200
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y)            00005300
         EXEC(CMD('EVISROUT EVIEI20B ') ROUTE(ALL *))                   00005400
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           00005500

*
* MERGE the following 3 statements if using FDR into your MAT       @19A
*
*
IF MSGID = 'DFS4167A'
    & TEXT = MESSAGE
    & DOMAINID = %AOFDOM%
    & HDRMTYPE ¬= '#'                                                   00006500
    THEN EXEC(CMD('OUTREP ') ROUTE(ONE %AOFOPWTORS%))
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
*
IF MSGID = 'DFS4164W' & TEXT = MESSAGE & DOMAINID = %AOFDOM%
    & HDRMTYPE ¬= '#'
   THEN EXEC(CMD('IMSFWM 'MESSAGE))
```

```
       DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*                                                                       00005600
IF (MSGID = 'DFS4190I')
    & TEXT = MESSAGE
    & DOMAINID = %AOFDOM%
    & HDRMTYPE = 'E'
   THEN
    EXEC(CMD('EVISROUT EVIDISCQ')
       ROUTE(ALL *))
       DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
 IF MSGID = 'DFS3410I'                                                  00005700
    & (HDRMTYPE = 'E' | HDRMTYPE = '"')                                 00005800
    & DOMAINID = %AOFDOM%                                               00005900
    & TEXT = MESSAGE                                                    00006000
    THEN EXEC(CMD('EVISROUT EVIEI00Q') ROUTE(ALL *))                    00006100
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           00006200
*                                                                       00006300
*   send ACTIVMSG to GSSOPER                                       @10A 00006300
 IF MSGID = 'DFS228I'                                                   00006400
    & (HDRMTYPE = 'E' | HDRMTYPE = '"')                                 00006500
    & DOMAINID = %AOFDOM%                                               00006600
    & TEXT = . 'DLS REGION INITIALIZATION COMPLETE' .                   00006700
    THEN EXEC(CMD('ACTIVMSG UP=YES') ROUTE(ONE %AOFOPGSSOPER%))         00006800
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           00006900
*                                                                       00007000
 IF MSGID = 'DFS0488I'                                                  00007100
    & (HDRMTYPE = 'E')                                                  00007200
    & (TOKEN(2) = 'UNLOCK' |                                            00007300
       TOKEN(3) = 'UNLOCK')                                             00007400
    & TEXT = . 'RC = 00' .                                              00007500
    & DOMAINID = %AOFDOM%                                               00007600
    THEN EXEC(CMD('EVISROUT EVIEO005 ') ROUTE(ALL *))                   00007700
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           00007800
*                                                                       00007900
 IF MSGID = 'DFS551I'                                                   00008000
    & DOMAINID = %AOFDOM%                                               00008100
    & (HDRMTYPE = 'E' | HDRMTYPE = '"')                                 00008200
    THEN EXEC(CMD('EVISROUT EVIES002 ') ROUTE(ALL *))                   00008300
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           00008400
 IF MSGID = 'DFS552I'                                                   00008500
    & DOMAINID = %AOFDOM%                                               00008600
    & (HDRMTYPE = 'E' | HDRMTYPE = '"')                                 00008700
    THEN EXEC(CMD('EVISROUT EVIES003 ') ROUTE(ALL *))                   00008800
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           00008900
*                                                                       00009000
*                                                             @13C @18C 00009100
IF MSGID = 'DFS554A'
   & (TEXT = . '(1)' .         |
      TEXT = . '(2)' .         |
      TEXT = . '(4)' . )
   & DOMAINID = %AOFDOM%
    & (HDRMTYPE = 'E' | HDRMTYPE = '"')
   & TEXT = MESSAGE
   THEN EXEC(CMD('EVIEY00S 'MESSAGE) ROUTE(ONE *))
             DISPLAY(N) NETLOG(Y) SYSLOG(Y);
*                                                                       00010300
IF MSGID = 'DFS554A'                                                    00010400
```

```
   & (TOKEN(7) = 'PROGRAM NAME')                                     00010500
   & (TEXT = . ')' . 'SMB' .  |                                      00010600
      TEXT = . ')' . 'PSB' .  |                                      00010700
      TEXT = . ')' . ',0456' . )                                     00010800
   & DOMAINID = %AOFDOM%                                             00010900
   & (HDRMTYPE = 'E' | HDRMTYPE = '"')                               00011000
   & TEXT = MESSAGE                                                  00011100
   THEN EXEC(CMD('EVIEY00S 'MESSAGE) ROUTE(ONE *))                   00011200
             DISPLAY(N) NETLOG(Y) SYSLOG(Y);                         00011300
*                                                        @06C @12C  00011400
 IF (MSGID = 'DFS0618A' | MSGID = 'DFS3626I'|
     MSGID = 'DFS166' | MSGID = 'DFS3131A')                          00011500
    & (HDRMTYPE = 'E' | HDRMTYPE = '"' | HDRMTYPE = '>')             00011600
    & DOMAINID = %AOFDOM%                                            00011700
    THEN EXEC(CMD('EVISROUT EVIEI006 ') ROUTE(ALL *))                00011800
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);        00011900
*                                                                    00012000
 IF MSGID = 'DFS616I'                                                00012100
    & (HDRMTYPE = 'E' | HDRMTYPE = '"')                              00012200
    & DOMAINID = %AOFDOM%                                            00012300
    THEN EXEC(CMD('EVISROUT EVIE0010 ') ROUTE(ALL *))                00012400
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);        00012500
*                                                                    00012600
 IF MSGID = 'DFS627I'                                                00012700
    & (HDRMTYPE = 'E' | HDRMTYPE = '"')                              00012800
    & DOMAINID = %AOFDOM%                                            00012900
    THEN EXEC(CMD('EVISROUT EVIER001 ') ROUTE(ALL *))                00013000
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);        00013100
*     remove check for CTL TCB                                 @11C  00013200
 IF MSGID = 'DFS629I'                                                00013300
    & (HDRMTYPE = 'E' | HDRMTYPE = '"')                              00013400
    & DOMAINID = %AOFDOM%                                            00013600
    & TEXT = . 'ABEND' .                                       @14A
    THEN EXEC(CMD('EVISROUT EVIER000 ') ROUTE(ALL *))                00013700
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);        00013800
*                                                        @17C @01C* 00013900
 IF MSGID = 'DFS810A'  & HDRMTYPE = '>'                              00014000
    & TEXT = MESSAGE                                                 00014100
    & DOMAINID = %AOFDOM%                                            00014200
    THEN EXEC(CMD('OUTREP ' MESSAGE) ROUTE(ONE %AOFOPWTORS%))        00014300
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y)         00014400
         EXEC(CMD('EVISROUT EVIEI00B ') ROUTE(ONE %AOFOPWTORS%))     00014500
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);        00014600
*                                                                    00014700
* Look for SHUTDOWN in either token 3 or 4.                    @07A  00016600
 IF MSGID = 'DFS994I'                                                00014800
   & (TOKEN(3) = 'SHUTDOWN' | TOKEN(4) = 'SHUTDOWN')                 00014900
   & DOMAINID = %AOFDOM%                                             00015000
    & (HDRMTYPE = 'E' | HDRMTYPE = '"')                              00015100
   THEN EXEC(CMD('EVISROUT EVIET006 ') ROUTE(ALL *))                 00015200
             DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);         00015300
 IF MSGID = 'DFS994I'                                                00015400
    & (TOKEN(3) = 'START' | TOKEN(4) = 'START')                      00015500
    & DOMAINID = %AOFDOM%                                            00015600
    & JOBNAME  = SVJOB                                               00015700
    & (HDRMTYPE = 'E' | HDRMTYPE = '"')                              00015800
    THEN EXEC(CMD('EVISROUT EVIEI00C ') ROUTE(ALL *))                00015900
         EXEC(CMD('EVISROUT EVIEC005 'SVJOB' INITIALIZE') ROUTE(ALL *)) 00016000
```

```
        EXEC(CMD('EVISROUT EVIECR04 'SVJOB) ROUTE(ALL *))              00016100
             DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           00016200
*                                                                      00016300
* Pre-IMS/ESA V5R1                                               @05A  00016400
*   DFS994I TAKEOVER COMPLETED. imsid                            @05A  00016500
*                                                                @05A  00016600
* IMS/ESA V5R1:                                                  @05A  00016700
*   DFS994I XRF TAKEOVER COMPLETED. imsid                        @05A  00016800
*                                                                @05A  00016900
 IF MSGID = 'DFS994I'                                                  00017000
   & ( TOKEN(2) = 'TAKEOVER' |                                         00017100
     ( TOKEN(2) = 'XRF' & TOKEN(3)='TAKEOVER') )                       00017200
   & DOMAINID = %AOFDOM%                                               00017300
   & JOBNAME  = SVJOB                                                  00017400
   & (HDRMTYPE = 'E' | HDRMTYPE = '"')                                 00017500
   THEN EXEC(CMD('EVISROUT EVIE0006 ') ROUTE(ALL *))                   00017600
        EXEC(CMD('EVISROUT EVIECO05 'SVJOB' INITIALIZE') ROUTE(ALL *)) 00017700
        EXEC(CMD('EVISROUT EVIECR04 'SVJOB) ROUTE(ALL *))              00017800
             DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           00017900
*                                                           @17C @01C* 00018000
 IF MSGID = 'DFS996I'  & HDRMTYPE='>'                                  00018100
   & TEXT = MESSAGE                                                    00018200
   & DOMAINID = %AOFDOM%                                               00018300
   THEN EXEC(CMD('OUTREP ' MESSAGE) ROUTE(ONE %AOFOPWTORS%))           00018400
             DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           00018500
*                                                           @17C @01C* 00018600
 IF MSGID = 'DFS972A'  & HDRMTYPE='>'                                  00018700
   & TEXT = MESSAGE                                                    00018800
   & DOMAINID = %AOFDOM%                                               00018900
   THEN EXEC(CMD('OUTREP ' MESSAGE) ROUTE(ONE %AOFOPWTORS%))           00019000
             DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           00019100
*                                                           @17C @01C* 00019200
 IF MSGID = 'DFS3139I' & HDRMTYPE='>'                                  00019300
   & TEXT = MESSAGE                                                    00019400
   & DOMAINID = %AOFDOM%                                               00019500
   THEN EXEC(CMD('OUTREP ' MESSAGE) ROUTE(ONE %AOFOPWTORS%))           00019600
             DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           00019700
*                                                                      00019800
*   send ACTIVMSG to GSSOPER                                     @10A  00006300
 IF MSGID = 'DFS3613I'                                                 00019900
   & (HDRMTYPE = 'E' | HDRMTYPE = '"')                                 00020000
   & DOMAINID = %AOFDOM%                                               00020100
   & TEXT = . 'DRC TCB INITIALIZATION COMPLETE' .                      00020200
   THEN EXEC(CMD('ACTIVMSG UP=YES') ROUTE(ONE %AOFOPGSSOPER%))         00006800
             DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           00020400
*
* add call to EVIEAOPS                                        @l1a
 IF MSGID = 'DFS3613I' & HDRMTYPE = 'E'
   & DOMAINID = %AOFDOM%
   & TEXT = . 'TCO TCB INITIALIZATION COMPLETE' .
   THEN EXEC(CMD('EVIEAOPS ') ROUTE(ALL *))
        EXEC(CMD('EVISROUT EVIEET00 ') ROUTE(ALL *))
             DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
* add call to EVIEAOPS                                        @l1a
 IF MSGID = 'DFS3613I' & HDRMTYPE = 'E'
   & DOMAINID = %AOFDOM%
   & TEXT = . 'DLG TCB INITIALIZATION COMPLETE' .
```

```
    THEN EXEC(CMD('EVIEAOPS ') ROUTE(ALL *))
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*                                                                @02A
 IF MSGID = 'DFS3343E' & HDRMTYPE = 'E'
    & DOMAINID = %AOFDOM%
    & TEXT = . 'UNABLE TO INITIALIZE TIME CONTROL OPTION'.
    THEN EXEC(CMD('EVISROUT EVIEET00 ') ROUTE(ALL *))
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*                                                                @02A
 IF (MSGID = 'DFS3350E'| MSGID = 'DFS3351E') & HDRMTYPE = 'E'
    & DOMAINID = %AOFDOM%
    & TEXT = . 'TCO ABNORMALLY TERMINATED'.
    THEN EXEC(CMD('EVISROUT EVIEET00 ') ROUTE(ALL *))
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
 IF MSGID = 'DFS3801A'
    & (HDRMTYPE = 'E' | HDRMTYPE = '"')
    & DOMAINID = %AOFDOM%
    THEN EXEC(CMD('EVISROUT EVIEI00D ') ROUTE(ALL *))
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
 IF MSGID = 'DFS3802W'
    & (HDRMTYPE = 'E' | HDRMTYPE = '"')
    & DOMAINID = %AOFDOM%
    THEN EXEC(CMD('EVISROUT EVIEI005 ') ROUTE(ALL *))
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
 IF MSGID = 'DFS3804I'
    & (HDRMTYPE = 'E' | HDRMTYPE = '"')
    & DOMAINID = %AOFDOM%
    THEN EXEC(CMD('EVISROUT EVIEI00A ') ROUTE(ALL *))
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
 IF MSGID = 'DFS3806A'
    & (HDRMTYPE = 'E' | HDRMTYPE = '"')
    & DOMAINID = %AOFDOM%
    THEN EXEC(CMD('EVISROUT EVIEI00F ') ROUTE(ALL *))
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
 IF MSGID = 'DFS3838I'
    & (HDRMTYPE = 'E' | HDRMTYPE = '"')
    & TEXT = MESSAGE
    & DOMAINID = %AOFDOM%
    THEN EXEC(CMD('EVISROUT EVIEI00G ') ROUTE(ALL *))
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
 IF MSGID = 'DFS3839I'
    & (HDRMTYPE = 'E' | HDRMTYPE = '"')
    & TEXT = MESSAGE
    & DOMAINID = %AOFDOM%
    THEN EXEC(CMD('EVISROUT EVIEI00G ') ROUTE(ALL *))
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*                                                             @17C @01C*
 IF MSGID = 'DFS3854A' & HDRMTYPE = '>'
    & DOMAINID = %AOFDOM%
    & TEXT = MESSAGE
    THEN EXEC(CMD('OUTREP ' MESSAGE) ROUTE(ONE %AOFOPWTORS%))
         EXEC(CMD('EVISROUT EVIE0004 ') ROUTE(ALL *))
```

```
               DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
 *
  IF MSGID = 'DFS3866A'
     & (HDRMTYPE = 'E' | HDRMTYPE = '"')
     & DOMAINID = %AOFDOM%
     THEN EXEC(CMD('EVISROUT EVIEI00F ') ROUTE(ALL *))
               DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
 *
  IF MSGID = 'DFS3869A'
     & (HDRMTYPE = 'E' | HDRMTYPE = '"')
     & DOMAINID = %AOFDOM%
     THEN EXEC(CMD('EVISROUT EVIEO001 ') ROUTE(ALL *))
               DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
 *
  IF MSGID = 'DFS3869I'
     & (HDRMTYPE = 'E' | HDRMTYPE = '"')
     & DOMAINID = %AOFDOM%
     THEN EXEC(CMD('EVISROUT EVIEO001 ') ROUTE(ALL *))
               DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
 *
  IF MSGID = 'DFS3873I'
     & (HDRMTYPE = 'E' | HDRMTYPE = '"')
     & DOMAINID = %AOFDOM%
     THEN EXEC(CMD('EVISROUT EVIEI008 ') ROUTE(ALL *))
               DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
 *
  IF (MSGID = 'DFS3872I' |
      MSGID = 'DFS3877I')
     & (HDRMTYPE = 'E' | HDRMTYPE = '"')
     & DOMAINID = %AOFDOM%
     THEN EXEC(CMD('EVISROUT EVIEI009 ') ROUTE(ALL *))
               DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
 *
  IF MSGID = 'DFS3890I'
     & (HDRMTYPE = 'E' )
     & (TOKEN(2) = 'TAKEOVER' | TOKEN(3) = 'TAKEOVER')
     & DOMAINID = %AOFDOM%
     THEN EXEC(CMD('EVISROUT EVIEO000 ') ROUTE(ALL *))
               DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
 *
  IF MSGID = 'DFS3891I'
     & (HDRMTYPE = 'E' )
     & (TOKEN(2) = 'TAKEOVER' | TOKEN(3) = 'TAKEOVER')
     & DOMAINID = %AOFDOM%
     THEN EXEC(CMD('EVISROUT EVIEO002 ') ROUTE(ALL *))
               DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
 *
  IF MSGID = 'DFS3898W'
     & (HDRMTYPE = 'E' | HDRMTYPE = '"')
     & DOMAINID = %AOFDOM%
     THEN EXEC(CMD('EVISROUT EVIEI005 ') ROUTE(ALL *))
               DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
 *
  IF MSGID = 'DFS3899W'
     & (HDRMTYPE = 'E' | HDRMTYPE = '"')
     & DOMAINID = %AOFDOM%
     THEN EXEC(CMD('EVISROUT EVIEI005 ') ROUTE(ALL *))
          DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
```

```
 IF MSGID='DFS0414I' & TOKEN(2) = 'PERMANENT'                          00032400
    & TEXT=MESSAGE                                                     00032500
   THEN EXEC(CMD('IMSFWM ' MESSAGE) ROUTE(ONE *)) ;                    00032600
*                                                                      00032700
 IF (MSGID='DFS0414I' | MSGID='DFS0739X' )                             00032800
      & TEXT=MESSAGE                                                   00032900
   THEN EXEC(CMD('IMSFWM 'MESSAGE) ROUTE(ONE *)) ;                     00033000
*                                                                      00033100
 IF (MSGID='DFS3258A' | MSGID='DFS3260I') & TEXT=MESSAGE               00033200
   THEN EXEC(CMD('EVIEY00S ' MESSAGE) ROUTE(ALL *))                    00033300
        EXEC(CMD('IMSFWM 'MESSAGE) ROUTE(ONE *)) ;                     00033400
*                                                                      00033500
 IF MSGID='DFS3257I' & TOKEN(5) = 'SWITCHED' & TOKEN(8) = 'DFSOLP'. &  00033600
    JOBNAME = SVJOB                                                    00033700
   THEN EXEC(CMD('EVISROUT EVIECO05 'SVJOB' SWITCH') ROUTE(ALL *));    00033800
*                                                                      00033900
 IF MSGID='DFS3257I' & TOKEN(2) = 'OLDS' & TOKEN(3) = 'DEALLOCATED' &  00034000
    TOKEN(5) = 'DFSOLP'. & JOBNAME = SVJOB                             00034100
   THEN EXEC(CMD('EVISROUT EVIECO05 'SVJOB) ROUTE(ALL *));             00034200
*                                                                      00034300
 IF MSGID='DFS3257I' & TEXT=MESSAGE                                    00034400
   THEN EXEC(CMD('IMSFWM 'MESSAGE) ROUTE(ALL *));                      00034500
*                                                                      00034600
* ADDED DFS2717, DFS2719I, DFS2721I                               @04A 00034700
* Add HDRMTYPE check.                                             @20A
 IF (MSGID  = 'DFS000I' | MSGID = 'DFS058I' | MSGID = 'DFS134I' |      00034800
     MSGID  = 'DFS063I' |                                              00034900
     MSGID  = 'DFS140I' | MSGID = 'DFS141I' | MSGID = 'DFS142I' |      00035000
     MSGID  = 'DFS2717' | MSGID = 'DFS2719I'| MSGID = 'DFS2721I')      00035100
    & (HDRMTYPE = 'E' | HDRMTYPE = '"' | HDRMTYPE = '>')
    & DOMAINID = %AOFDOM%                                              00035200
   THEN   EXEC(CMD('EVISROUT') ROUTE(ONE *));                          00035300
*                                                                      00035400
 IF (MSGID = 'DFS2160I' | MSGID = 'DFS2161I' | MSGID = 'DFS2168I' |    00035500
     MSGID = 'DFS2169I' | MSGID = 'DFS2142'  | MSGID = 'DFS2140'  |    00035600
     MSGID = 'DFS2236')                                                00035700
    & TEXT = MESSAGE                                                   00035800
    THEN EXEC(CMD('EVIEY00S 'MESSAGE) ROUTE(ALL *))                    00035900
              DISPLAY(N) NETLOG(Y) SYSLOG(Y);                          00036000
*                                                                      00036100
IF MSGID='DFS2500I' & TOKEN(3) = 'DFSOLP'. & TOKEN(5) = 'ALLOCATED' &  00036200
   JOBNAME = SVJOB                                                     00036300
  THEN EXEC(CMD('EVISROUT EVIECO05 'SVJOB) ROUTE(ALL *));              00036400
*                                                                8@08A 00036500
***********************************************************************00036500
* IRLM-RELATED IMS MESSAGES                                            00036500
***********************************************************************00036500
 IF (MSGID = 'DFS626I' | MSGID = 'DFS2011I' | MSGID = 'DFS2012I')      00036500
     & TEXT=MESSAGE                                                    00036500
    THEN EXEC(CMD('IMSFWM 'MESSAGE) ROUTE(ALL *))                      00036500
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);          00036500

**************************************************************** @16A* 00386240
*                                                                @16A* 00387230
* Trap unknown replies                                           @16A* 00388220
*                                                                @16A* 00389210
*************************************************************** 4@16A* 00390200
                                                                       00391190
```

```
      IF IFRAUWF1(6) = '1' & DOMAINID = %AOFDOM%                         00392180
         THEN EXEC(CMD('OUTREP ')ROUTE(ONE %AOFOPWTORS%));               00393170
                                                                         00036600
   ALWAYS;                                                               00036700
  END;                                                                   00036800
 *                                                                       00036900
```

EVIMDSP1

```
************************************************************************00010000
* (C) COPYRIGHT IBM CORP. 1995 - ALL RIGHTS RESERVED                   *00020000
************************************************************************00030000
*                                                                      *00040000
*        Licensed materials - Property of IBM                          *00050000
*        Restricted materials of IBM                                   *00060000
*        5685-151                                                      *00070000
*        (C) Copyright IBM Corp. 1995                                  *00080000
*                                                                      *00090000
*        US government users restricted rights -                       *00100000
*        use, duplication or disclosure restricted by                  *00110000
*        GSA ADP schedule contract with IBM corp.                      *00120000
*                                                                      *00130000
************************************************************************00140000
*   APAR#     DATE                                                     *00150000
*  -------  --------                                                   *00160000
************************************************************************00170000
* Description:  IMS message table for the DSP* messages                *00180000
************************************************************************00190000
****IMS*****************************************************************00200000
*  MESSAGE TABLE ENTRIES FOR IMS PO RECON RECOVERY AREA.                00210000
************************************************************************00220000
 IF MSGID = 'DSP' .                                                     00230000
   THEN                                                                 00240000
     BEGIN;                                                             00250000
*                                                                       00260000
IF MSGID='DSP0381'. &                                                   00270000
   JOBNAME = SVJOB                                                      00280000
  THEN EXEC(CMD('EVIECR04 'SVJOB) ROUTE(ALL *));                        00290000
*                                                                       00300000
  ALWAYS;                                                               00310000
 END;                                                                   00320000
*                                                                       00330000
```

EVIMDXR1

```
***********************************************************************00030000
*                                                                     *00040000
*        Licensed materials - Property of IBM                         *00050000
*        5685-151                                                     *00070000
*        (C) Copyright IBM Corp. 1995, 1996                           *00080001
*                                                                     *00090000
***********************************************************************00140000
*     APAR#      Date   By Description                                *00150000
*  ----------- -------- -- ------------------------------------------ *00160000
*  $02=OW19688 04/05/96 CG SAMPLE EVIMDXR1 CONTAINS BEGIN END BLOCK FOR*
*                          DXR BUT WITHIN THE BLOCK THERE ARE COMPARES *
*                          FOR MESSAGE IDS BEGINNING WITH DFS.         *
*                                                                      *
*  $01=OW17371 12/08/95 CG IMS APPLICATION GOES FROM UP TO ACTIVE TO   *
*                          STARTED2. (REMOVE 2ND DXR009I)              *
*                                                                      *
***********************************************************************00170000
* Description:  IMS message table for the DXR* messages              *00180000
***********************************************************************00190000
 IF MSGID = 'DXR' .                                                    00200000
   THEN                                                                00210000
     BEGIN;                                                            00220000
*                                                                 @01C 00230000
 IF MSGID = 'DXR009I'                                                  00240000
    & DOMAINID = %AOFDOM%                                              00250000
    & TEXT = . 'INITIALIZATION COMPLETE' .                             00260000
    THEN EXEC(CMD('AOFRSGUP ')  ROUTE(ONE %AOFOPGSSOPER%))             00270000
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);          00280000
*                                                                      00290000
***********************************************************************00300000
* IRLM MESSAGES                                                        00310000
***********************************************************************00320000
 IF (MSGID = 'DXR008E' | MSGID = 'DXR017E'  | MSGID = 'DXR019E'  |     00330000
     MSGID = 'DXR021E' | MSGID = 'DXR022E'  | MSGID = 'DXR023E'  |     00340000
     MSGID = 'DXR031E' | MSGID = 'DXR027I'  | MSGID = 'DXR025I'  |     00350000
     MSGID = 'DXR016E')                                                00360000
     & TEXT=MESSAGE                                                    00370000
    THEN EXEC(CMD('IMSFWM 'MESSAGE) ROUTE(ALL *))                      00380000
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);          00390000
*                                                                      00400000
 IF (MSGID = 'DXR002I' | MSGID = 'DXR007E'  | MSGID = 'DXR011I'  |     00410000
     MSGID = 'DXR018E' | MSGID = 'DXR030I'  | MSGID = 'DXR034I'  |     00420000
     MSGID = 'DXR035I' | MSGID = 'DXR045W')                            00430000
     & TEXT=MESSAGE                                                    00440000
    THEN EXEC(CMD('IMSFWM 'MESSAGE) ROUTE(ALL *))                      00450000
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);          00460000
*                                                               9@01D  00560000
 ALWAYS;                                                               00570000
END;                                                                   00580000
```

EVIMCQS1

```
***********************************************************************
* (C) COPYRIGHT IBM CORP. 1999 - ALL RIGHTS RESERVED                  *
***********************************************************************
*                                                                     *
*        LICENSED MATERIALS - PROPERTY OF IBM                         *
*        RESTRICTED MATERIALS OF IBM                                  *
*        5685-151                                                     *
*        (C) COPYRIGHT IBM CORP. 1999                                 *
*                                                                     *
*        US GOVERNMENT USERS RESTRICTED RIGHTS -                      *
*        USE, DUPLICATION OR DISCLOSURE RESTRICTED BY                 *
*        GSA ADP SCHEDULE CONTRACT WITH IBM CORP.                     *
*                                                                     *
***********************************************************************
*     APAR#      DATE   BY DESCRIPTION                                *
*  ----------- -------- -- ------------------------------------------*
*  $L1=OW35606,V1R4,20JAN99,APC(NS):CQS SUPPORT.                      *
*                                                                     *
***********************************************************************
* IMS CQS MESSAGES :EVIMCQS1                                          *
*                                                                     *
***********************************************************************
  IF MSGID = 'CQS' .
    THEN
      BEGIN;
*
  IF (MSGID = 'CQS0031A' |
      MSGID = 'CQS0032A' |
      MSGID = 'CQS0033A')
    & TEXT = MESSAGE & DOMAINID = %AOFDOM%
     THEN
      EXEC(CMD('ISSUEREP AUTOTYP=RESTART')
         ROUTE(ONE %AOFOPWTORS%));
*
  IF MSGID='CQS0020I' & DOMAINID = %AOFDOM%
     THEN EXEC(CMD('ACTIVMSG UP=YES') ROUTE(ONE %AOFOPGSSOPER%))
          DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
  IF MSGID='CQS0034A' & TEXT = MESSAGE
     THEN EXEC(CMD('IMSFWM 'MESSAGE))
          DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
  IF MSGID='CQS0205E' & TEXT = MESSAGE & TOKEN(3) = STRUCT
     & JOBNAME = JOB
     THEN EXEC(CMD('EVISTRCT 'MESSAGE))
          DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
  IF MSGID='CQS0206I' & TEXT = MESSAGE & TOKEN(3) = STRUCT
     THEN EXEC(CMD('EVISTRMN ' STRUCT))
          DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
  IF (MSGID='CQS0008W' |
      MSGID='CQS0009W')
    & TEXT = MESSAGE
     THEN EXEC(CMD('IMSFWM 'MESSAGE))
          DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
  IF (MSGID='CQS0240I' |
      MSGID='CQS0242E' |
      MSGID='CQS0260I' |
      MSGID='CQS0263E')
    & TEXT = MESSAGE
```

```
      THEN EXEC(CMD('IMSFWM 'MESSAGE))
           DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);

    ALWAYS;
   END;
```

EVIMEVI1

```
*********************************************************************00030000
*                                                                   *00040000
*       Licensed materials - Property of IBM                        *00050000
*       5685-151                                                    *00070000
*       (C) Copyright IBM Corp. 1996                                *00080007
*                                                                   *00090000
*********************************************************************00140000
*    APAR#      Date   By Description                               *00150007
* ----------- -------- -- ----------------------------------------- *00160007
* $04=OW35606,V1R4,07Jan99,APC(NS):CQS support.                     *
*                                                                   *
* $03=OW27627,V1R4,10Jul97,APC(AE): No change - PE OW23923          *
*                                                                   *
* $02=OW23923 01/15/97 BW DFS996I WTOR REPLIES ISSUED AND EATEN UP  *
*                         WHEN AOI INITIALIZATION HAS NOT COMPLETED *
*                                                                   *
* $01=OW20827 06/05/96 CG PTF UW90273 DID NOT INCLUDE ++HOLD ACTION *
*                         STATEMENTS INSTALLER MAY MISS NECESSARY   *
*                         INSTALLATION INFORMATION                  *
*                                                                   *
* $L1=OW19977 05/31/96 LK AOC IMS-AUTOMATION FEATURE SUPPORT FOR IMS *
*                         5.1 ARM RECOVERY                          *
*                                                                   *
*********************************************************************00180000
* Description:  IMS message table for the EVI* messages             *00190000
*********************************************************************00200000
 IF MSGID = 'EVI' .                                                  00210000
   THEN                                                              00220000
     BEGIN;                                                          00230000
*                                                                    00240000
*  CALL EVIEI00L TO UPDATE TASK VARS SUBSDIS TO "UP"             @02A 00250000
 IF MSGID = 'EVI200I'                                                00260000
   & TEXT = MESSAGE                                                  00270000
   & DOMAINID = %AOFDOM%                                             00280000
   THEN EXEC (CMD('EVIE200I 'MESSAGE) ROUTE(ALL *))                  00290000
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(N)         00300000
        EXEC (CMD('EVISROUT EVIEI00L') ROUTE(ALL *))                 00290000
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);        00300000
*
*   If using shared queues MERGE the following message EXEC command@04A
*   into your MAT this will allow the /CQSET command to be issued  @04A
*                                                                @04C 00310000
*    EXEC (CMD('EVIDISCQ 'MESSAGE ) ROUTE(ONE *))                @04C 00310000
*          DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(N);       @04C 00310000
*                                                                @04C 00310000
 IF MSGID = 'EVI146I'                                                00320000
   & TOKEN(11) = SVJOB                                               00330000
   & TEXT = MESSAGE                                                  00340000
   & DOMAINID = %AOFDOM%                                             00350000
   THEN EXEC (CMD('EVISROUT JOBNAME='SVJOB' EVIEI00H 'MESSAGE)       00360000
                   ROUTE(ALL *))                                     00370000
        EXEC (CMD('EVIDISCQ 'MESSAGE ) ROUTE(ONE *))
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(N);        00380000
*
*   If using shared queues MERGE the following message trap entry  @04A
*   into your MAT this will allow the /CQSET command to be issued  @04A
*
IF (MSGID = 'EVI130I')
  & DOMAINID = %AOFDOM%
  THEN EXEC(CMD('EVIDISCQ ') ROUTE(ONE *))
       EXEC(CMD('ACTIVMSG UP=YES') ROUTE(ALL *))
```

```
             DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(N);
*                                                                       00390000
 IF MSGID = 'EVI' . 'A' | MSGID = 'EVI' . 'E' THEN                      00400000
    HOLD(Y) DISPLAY(Y);                                                 00410000
*                                                                       00420000
************************************************************************00430002
* ARM Notify FP of contents of EVIALL table                         @l1a00440002
************************************************************************00450002
*                                                                       00460002
* CROSS-SYSTEM MESSAGE ROUTING                                          00470000
 IF MSGID  = 'EVI894I' & TEXT = MESSAGE                                 00480008
   THEN EXEC( CMD('EVIEAAST FP 'MESSAGE) ROUTE(ONE *))                  00490002
        DISPLAY(Y) NETLOG(N);                                           00500000
*                                                                       00510002
************************************************************************00520002
* IMS SINGLE POINT OF CONTROL SUPPORT MESSAGES                          00530002
************************************************************************00540002
*                                                                       00550002
* CROSS-SYSTEM MESSAGE ROUTING                                          00560002
 IF MSGID  = 'EVI697'                                                   00570002
   THEN EXEC( CMD('EVISX003 ') ROUTE(ONE *))                            00580002
        DISPLAY(Y) NETLOG(N);                                           00590002
*                                                                       00600000
* SINGLE POINT OF CONTROL                                               00610000
 IF MSGID  = 'EVI895I' & TEXT = MESSAGE                                 00620000
   THEN EXEC( CMD('EVISXDMB 'MESSAGE) ROUTE(ONE *))                     00630000
        DISPLAY(Y) NETLOG(N);                                           00640000
*                                                                       00650000
  ALWAYS;                                                               00660000
 END;                                                                   00670000
```

EVIMIOS1

```
************************************************************************
* (C) COPYRIGHT IBM CORP. 1995 - ALL RIGHTS RESERVED                   *
************************************************************************
*                                                                      *
*        Licensed materials - Property of IBM                          *
*        Restricted materials of IBM                                   *
*        5685-151                                                      *
*        (C) Copyright IBM Corp. 1995                                  *
*                                                                      *
*        US government users restricted rights -                       *
*        use, duplication or disclosure restricted by                  *
*        GSA ADP schedule contract with IBM corp.                      *
*                                                                      *
************************************************************************
*   APAR#     DATE                                                     *
*  -------  --------                                                   *
************************************************************************
* Description:  IMS message table for the IOS* messages                *
************************************************************************
 IF MSGID = 'IOS' .
   THEN
     BEGIN;
*
 IF MSGID = 'IOS071E'
    & DOMAINID = %AOFDOM%
    & TEXT = . ',' . ',' SVJOB ',' .
    THEN EXEC(CMD('EVISROUT JOBNAME='SVJOB' EVIE0008 ' SVJOB)
     ROUTE(ALL *))
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
 IF MSGID = 'IOS071I'
    & DOMAINID = %AOFDOM%
    & TEXT = . ',' . ',' SVJOB ',' .
    THEN EXEC(CMD('EVISROUT JOBNAME='SVJOB' EVIE0008 ' SVJOB)
     ROUTE(ALL *))
              DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
  ALWAYS;
 END;
```

NetView Operator Definition

EVIOPF

```
***********************************************************************
* (C) COPYRIGHT IBM CORP. 1990, 1995 - ALL RIGHTS RESERVED            *
***********************************************************************
*                                                                     *
*        LICENSED MATERIALS - PROPERTY OF IBM                         *
*        RESTRICTED MATERIALS OF IBM                                  *
*        5685-151                                                     *
*        (C) COPYRIGHT IBM CORP. 1990, 1995                           *
*                                                                     *
*        US GOVERNMENT USERS RESTRICTED RIGHTS -                      *
*        USE, DUPLICATION OR DISCLOSURE RESTRICTED BY                 *
*        GSA ADP SCHEDULE CONTRACT WITH IBM CORP.                     *
*                                                                     *
***********************************************************************
*   APAR#     DATE                                                    *
*  -------  --------                                                  *
***********************************************************************
* CATEGORY       :  IMS AUTOMATION OPTION SETUP                       *
* DESCRIPTION    :  ADDITIONAL AUTOTASK ENTRIES FOR THE IMS AUTOMATION *
*                :  OPTION                                            *
*                :  COPY INTO NETVIEW'S DSIOPF MEMBER.                *
*                :                                                    *
***********************************************************************
*
AUTIMS       OPERATOR  PASSWORD=AUTIMS
             PROFILEN  EVIPRFAO
AUTIPPI      OPERATOR  PASSWORD=AUTIPPI
             PROFILEN  EVIPRFAO
AUTSURV      OPERATOR  PASSWORD=AUTSURV
             PROFILEN  EVIPRFAO
AUTIMSZ      OPERATOR  PASSWORD=AUTIMSZ
             PROFILEN  EVIPRFAO
AUTIMSA      OPERATOR  PASSWORD=AUTIMSA
             PROFILEN  EVIPRFAO
AUTIMSB      OPERATOR  PASSWORD=AUTIMSB
             PROFILEN  EVIPRFAO
AUTIMSC      OPERATOR  PASSWORD=AUTIMSC
             PROFILEN  EVIPRFAO
*
***********************************************************************
*  END OF IMS AUTOMATION OPTION AUTOTASK IDS
***********************************************************************
```

Required Command Model Entries

EVICMD

```
*********************************************************************
*********************************************************************
*                                                                   *
*        Licensed Materials - Property of IBM                       *
*        5685-151                                                   *
*        (C) Copyright IBM CORP. 1990, 1996                         *
*                                                                   *
*********************************************************************
*     APAR#      Date    By Description                             *
*  ----------- -------- -- ------------------------------------------ *
* $03=OW30099,V1R4,30Oct97,APC(AE): EVIESECUR needs to be included  *
*                                   after EVISX001.                 *
*                                                                   *
*  $02=OW20827 06/05/96 CG PTF UW90273 DID NOT INCLUDE ++HOLD ACTION  *
*                         STATEMENTS INSTALLER MAY MISS NECESSARY   *
*                         INSTALLATION INFORMATION                  *
*                                                                   *
*  $L1=OW19977 05/31/96 LK AOC IMS-AUTOMATION FEATURE SUPPORT FOR IMS *
*                         5.1 ARM RECOVERY                          *
*                                                                   *
*  $01=OW17438 12/13/95 CG AOC/IMS EVIEE000 STOPS TASK AOFTSTS DURING *
*                         INITIALIZATION.  ALSO DELETE DUPE EVISY001. *
*                                                                   *
*  PN27152  01/13/93   MP  IMSCMD OR EVIED011 CLIST DOES NOT ALLOW FOR *
*                         /CHE SNAPQ TO BE ISS UED ACROSS THE AOI.  *
*                                                                   *
*  PN26092  08/26/92   BC  MSGDSI002I INVALID COMMAND RECEIVED WHEN  *
*                         USER ISSUES IMSRCMD AS DOCUMENTED IN THE  *
*                         IMSAO PROGRAMMER'S REFERENCE              *
*                                                                   *
*  PN08478  11/01/91   RR  IMSPOST COMMAND FOR IMSAO ERRORS         *
*                                                                   *
*********************************************************************
* CATEGORY       :  IMS AUTOMATION OPTION SETUP                     *
* DESCRIPTION    :  ADDITIONAL COMMAND MODEL STATEMENTS FOR THE IMS  *
*                :  AUTOMATION OPTION                               *
*                :  COPY INTO NETVIEW'S DSICMD MEMBER.              *
*                :                                                  *
*********************************************************************
* Update DF routine
EVIED001   CMDMDL   MOD=DSICCP,ECHO=N
           CMDSYN   IMSDF
* Forward and Display IMS critical messages
EVIED003   CMDMDL   MOD=DSICCP,ECHO=N
           CMDSYN   IMSFWM
*
EVIEE00A   CMDMDL   MOD=DSICCP,ECHO=N
           CMDSYN   SYNCIMS
* POST external events         added PN08478
EVIEE200   CMDMDL   MOD=DSICCP,ECHO=N
           CMDSYN   IMSPOST
* service period timed startup  (timer uo)
EVIEI150   CMDMDL   MOD=DSICCP,ECHO=N
```

```
          CMDSYN   IMSTIMEU
* service period timed shutdown (timer down)
EVIET150   CMDMDL   MOD=DSICCP,ECHO=N
           CMDSYN   IMSTIMED
* operator line mode - startup
EVIEI160   CMDMDL   MOD=DSICCP,ECHO=N
           CMDSYN   STARTIMS
* operator line mode - shutdown
EVIET160   CMDMDL   MOD=DSICCP,ECHO=N
           CMDSYN   SHUTIMS
********************************************************************* *
EVISTS     CMDMDL   MOD=EVISCSTS,TYPE=R,RES=Y,ECHO=Y
EVIASTS    CMDMDL   MOD=EVISMSTS,TYPE=D,PARSE=N,RES=Y
* Service periods
EVISSSRV   CMDMDL   MOD=EVISSSRV,TYPE=R,PARSE=Y,RES=Y
EVISXDAT   CMDMDL   MOD=EVISXDAT,TYPE=R,PARSE=Y,RES=Y
* message reroutind
EVISROUT   CMDMDL   MOD=EVISROUT,TYPE=R,PARSE=N,RES=Y
* Global access
EVISXGBL   CMDMDL   MOD=EVISXGBL,TYPE=R,PARSE=N,RES=Y
* TRIGGERS
EVISX004   CMDMDL   MOD=EVISX004,TYPE=R,PARSE=Y,RES=Y
* SINGLE POINT OF CONTROL (SPOC)
*  Domain table lookup
EVISX022   CMDMDL   MOD=EVISX022,TYPE=R,PARSE=Y,RES=Y
           CMDSYN   IMSQRY
*  Domain table build
EVISXDMB   CMDMDL   MOD=EVISXDMB,TYPE=R,PARSE=Y,RES=Y
*  Domain table delete
EVISXDMD   CMDMDL   MOD=EVISXDMD,TYPE=R,PARSE=Y,RES=Y
*  Domain table rebuild
EVISXDMR   CMDMDL   MOD=EVISXDMR,TYPE=R,PARSE=Y,RES=Y
* Member read
EVISY001   CMDMDL   MOD=EVISY001,TYPE=R,PARSE=Y,RES=Y
* Tracing
EVISTRAC   CMDMDL   MOD=EVISTRAC,TYPE=R,PARSE=Y,RES=Y
           CMDSYN   EVITRACE
* Program-to-Program Interface
EVISNBRE   CMDMDL   MOD=EVISNBRE,TYPE=R,PARSE=N,RES=Y
EVISNCCI   CMDMDL   MOD=EVISNCCI,TYPE=R,PARSE=N,RES=Y
EVISNRSP   CMDMDL   MOD=EVISNRSP,TYPE=R,PARSE=N,RES=Y
EVISNMSG   CMDMDL   MOD=EVISNMSG,TYPE=R,PARSE=N,RES=Y
EVISNACK   CMDMDL   MOD=EVISNACK,TYPE=R,PARSE=N,RES=Y
* Message BUILD/ROUTE
EVISX002   CMDMDL   MOD=EVISX002,TYPE=R,PARSE=Y,RES=Y
           CMDSYN   IMSBMSG
EVISX003   CMDMDL   MOD=EVISX003,TYPE=R,PARSE=Y,RES=Y
*                                                                2@01D
* IMS operator command interface
EVIED011   CMDMDL   MOD=DSICCP,ECHO=N
           CMDSYN   IMSCMD
* IMS operator Main Menu
EVIE0000   CMDMDL   MOD=DSICCP,ECHO=N
           CMDSYN   IMS
*
EVIEX017   CMDMDL   MOD=DSICCP,ECHO=N
```

```
          CMDSYN   IMSRCMD
*  Security checking
EVISX001   CMDMDL   MOD=EVISX001,TYPE=R,PARSE=Y,RES=Y
           CMDSYN   IMSSEC
           CMDCLASS 1,2
*                                                              3@03M
*          Include IMS Automation security checking parameters
%INCLUDE EVISECUR
* ADD ALL SUBSYSTEM TABLE MODULE FOR PERFORMANCE           @L1a
EVIEAAST   CMDMDL   MOD=DSICCP,ECHO=N
*
* ADD ALL SUBSYSTEM TABLE MODULE FOR PERFORMANCE           @L1a
EVIEASTS   CMDMDL   MOD=DSICCP,ECHO=N
*
***********************************************************************
*  END OF IMS AUTOMATION COMMAND MODEL STATEMENTS
***********************************************************************
*
```

EVISECUR

```
*********************************************************************
* (C) COPYRIGHT IBM CORP. 1995 - ALL RIGHTS RESERVED                *
*********************************************************************
*                                                                   *
*        Licensed materials - Property of IBM                       *
*        Restricted materials of IBM                                *
*        5685-151                                                   *
*        (C) Copyright IBM Corp. 1995                               *
*                                                                   *
*        US government users restricted rights -                    *
*        use, duplication or disclosure restricted by               *
*        GSA ADP schedule contract with IBM corp.                   *
*                                                                   *
*********************************************************************
*   APAR#     DATE                                                  *
*  -------  --------                                                *
*********************************************************************
*  DESCRIPTION: IMS SECURITY CHECKING                               *
*********************************************************************
*  NOTES:                                                           *
*  KEYCLASS can be an IMS subsystem name (as defined to AOC/MVS), a *
*  domain, or group name.  VALCLASS identifies functions allowed for *
*  the subsystem, group or domain specified with the previous       *
*  KEYCLASS entry.  All IMS subsystems must have security defintions *
*  in DSICMD.                                                       *
*                                                                   *
*********************************************************************
IMS10Z     KEYCLASS 1                    IMS SUBSYSTEM
INQUIRY    VALCLASS 2                    OPERATOR INVOKED INQUIRY
STARTUP    VALCLASS 2                    OPERATOR INVOKED STARTUP
SHUTDOWN   VALCLASS 2                    OPERATOR INVOKED SHUTDOWN
TRIGGERS   VALCLASS 2                    TRIGGER/EVENT PROCESSING
SERVICEP   VALCLASS 2                    SERVICE PERIOD PROCESSING
SUPPORT    VALCLASS 2                    TRACE, MESSAGE FUNCTIONS
EVIPPICT   VALCLASS 2                    START, STOP PPI INTERFACE
IMSCMD     VALCLASS 2                    ANY IMSCMD FUNCTION
/DIS       VALCLASS 2                    IMSCMD /DISPLAY
/CHE       VALCLASS 2                    IMSCMD /CHECKPOINT
/FOR       VALCLASS 2                    IMSCMD /FORMAT
=OTHER     VALCLASS 2                    ANY FUNCTION NOT SPECIFIED
*********************************************************************
*          ANY OTHER IMS SUBSYSTEM/COMMAND                          *
*********************************************************************
=OTHER     KEYCLASS 2                    ANY OTHER IMS SUBSYSTEM
=OTHER     VALCLASS 2                    ANY FUNCTION NOT SPECIFIED
```

Sample Configuration Tables

EVIDMN

```
**********************************************************************
*                                                                    *
*        LICENSED MATERIALS - PROPERTY OF IBM                        *
*        5685-151                                                    *
*        (C) COPYRIGHT IBM CORP. 1990, 1996                          *
*                                                                    *
**********************************************************************
*    APAR#      DATE   BY DESCRIPTION                                *
* ----------- -------- -- ------------------------------------------ *
* $L1=OWXXXXX 05/31/96 CG ARM SPE FOR IMS                            *
*                                                                    *
**********************************************************************
* CATEGORY       :  IMS AUTOMATION OPTION SETUP                      *
* DESCRIPTION    :  ADDITIONAL TASK STATEMENTS FOR THE IMS           *
*                :  AUTOMATION OPTION                                *
*                :  COPY INTO NETVIEW'S DSIDNM MEMBER.               *
*                :  (SEE CHAPTER 9, STEP 14 OF "AOC/MVS IMS          *
*                :   PROGRAMMER'S REFERENCE AND INSTALLATION GUIDE",  *
*                :   SC23-3817).  ALSO, SEE SAMPLE BELOW.            *
**********************************************************************
*
*
*  ENABLE IAO PPI INTERFACE
*
        TASK   MOD=EVISNPPI,TSKID=EVINTASK,MEM=EVINTASK,PRI=5
*
* FOR ALTERNATE XRF DOMAINS, ADD THIS SAMPLE
*
*       TASK   MOD=DSIZDST,TSKID=AOFdomid,MEM=STSdomid,PRI=5
*
* END OF IMS ENTRIES - ABOVE ENTRY COMMENTED OUT, MAY BE CODED ABOVE
*
```

EVISTSM

```
**********************************************************************
*                                                                    *
*        LICENSED MATERIALS - PROPERTY OF IBM                        *
*        5685-151                                                    *
*        (C) COPYRIGHT IBM CORP. 1996                                *
*                                                                    *
**********************************************************************
* DSTINIT MEMBER FOR AOC IMS AUTOMATION XRF STATUS FILE              *
* DEFINES STATUS FILE IN OTHER DOMAIN                                *
*                                                                    *
* STSdomid - NAME OF THIS MEMBER WHEN SPECIFIC INSTANCE IS ESTABLISHED *
*            FROM THIS TEMPLATE.                                     *
*                                                                    *
* AOFSISTS - AOC/MVS STATUS-FILE RECORD INITIALIZATION EXIT.         *
*                                                                    *
* AOFdomid - DDNAME OF STATUS DATASET IN DOMAIN=domid.               *
*                                                                    *
**********************************************************************
  DSTINIT XITVN=AOFSISTS
  DSTINIT PDDNM=AOFdomid
  DSTINIT FUNCT=VSAM
  DSTINIT DSRBO=1
```

EVIS2CFG

```
***********************************************************************
* (C) COPYRIGHT IBM CORP. 1990, 1995 - ALL RIGHTS RESERVED            *
***********************************************************************
*                                                                     *
*        LICENSED MATERIALS - PROPERTY OF IBM                         *
*        RESTRICTED MATERIALS OF IBM                                  *
*        5685-151                                                     *
*        (C) COPYRIGHT IBM CORP. 1990, 1995                           *
*                                                                     *
*        US GOVERNMENT USERS RESTRICTED RIGHTS -                      *
*        USE, DUPLICATION OR DISCLOSURE RESTRICTED BY                 *
*        GSA ADP SCHEDULE CONTRACT WITH IBM CORP.                     *
*                                                                     *
***********************************************************************
* DESCRIPTION: AOC/MVS CONFIGURATION CONTROL FILE FOR IMS             *
*              INCLUDES ALL OTHER CONTROL FILE MEMBERS FOR IMS        *
***********************************************************************
*   APAR#     DATE                                                    *
*  -------  --------                                                  *
*$01=OW35606,V1R4,20Jan98,APC(NS):New includes for shared queue support*
***********************************************************************
%INCLUDE EVIS2STA              * INCLUDE THE STATE ACTION SAMPLES
%INCLUDE EVIS2OPS              * INCLUDE THE AUTOOPS SAMPLES
%INCLUDE EVIS2IMS              * INCLUDE THE NON-XRF REGION SAMPLES
%INCLUDE EVIS2DBC              * INCLUDE THE DBCTL REGION SAMPLES
%INCLUDE EVIS2XRF              * INCLUDE THE XRF REGION SAMPLES
%INCLUDE EVIS2CRT              * INCLUDE THE DDF/SDF SAMPLES
%INCLUDE EVIS2LCL              * INCLUDE THE RESIDENT SAMPLES
* If using FDR uncomment out the following 2 includes          @01A
*%INCLUDE EVIS2SPX              * INCLUDE THE SHARED QUEUES IMS
*%INCLUDE EVIS2FDR              * INCLUDE THE FDR SAMPLES
* If using shared queues uncomment out the following include    @01A
*%INCLUDE EVIS2CQS              * INCLUDE THE CQS SAMPLES
```

EVIS2STA

```
**********************************************************************
* (C) COPYRIGHT IBM CORP. 1990, 1995 - ALL RIGHTS RESERVED           *
**********************************************************************
*                                                                    *
*        LICENSED MATERIALS - PROPERTY OF IBM                        *
*        RESTRICTED MATERIALS OF IBM                                 *
*        5685-151                                                    *
*        (C) COPYRIGHT IBM CORP. 1990, 1995                          *
*                                                                    *
*        US GOVERNMENT USERS RESTRICTED RIGHTS -                     *
*        USE, DUPLICATION OR DISCLOSURE RESTRICTED BY                *
*        GSA ADP SCHEDULE CONTRACT WITH IBM CORP.                    *
*                                                                    *
**********************************************************************
*   APAR#     DATE                                                   *
*  -------  --------                                                 *
*  $01=PN36820 08/17/93 MP IMSAO STATE/ACTION TABLE UPDATES AND      *
*                          CORRECTIONS.                              *
*                                                                    *
**********************************************************************
*
PRODUCT        IMS,SUBSYS=(IMS10Z,AREA=IMS),
                   SUBSYS=(IMS10AA,AREA=IMS),
                   SUBSYS=(IMS10AB,AREA=IMS)
***********        SUBSYS=(IMS10AC,AREA=IMSTEST)
*
AREA           IMS,MSC=EVISS002,
                   OLDS=EVISS003,
                   TRAN=EVISS005
*
*  TABLES EVISS10X ARE ONLY HERE IS DESCRIBE THE FLEXIBILTY OF USES
*    TEST TABLES FOR DIFFERRENT IMS'S
*
*AREA          IMSTEST,MSC=EVISS102,
*                   OLDS=EVISS103,
*                   TRAN=EVISS105
```

EVIS2OPS

```
*********************************************************************
* (C) COPYRIGHT IBM CORP. 1990, 1995 - ALL RIGHTS RESERVED          *
*********************************************************************
*                                                                   *
*        LICENSED MATERIALS - PROPERTY OF IBM                       *
*        RESTRICTED MATERIALS OF IBM                                *
*        5685-151                                                   *
*        (C) COPYRIGHT IBM CORP. 1990, 1995                         *
*                                                                   *
*        US GOVERNMENT USERS RESTRICTED RIGHTS -                     *
*        USE, DUPLICATION OR DISCLOSURE RESTRICTED BY               *
*        GSA ADP SCHEDULE CONTRACT WITH IBM CORP.                   *
*                                                                   *
*********************************************************************
* DESCRIPTION: AOC/MVS CONFIGURATION CONTROL FILE FOR IMS           *
*              DEFINE ALL IMS AUTOOPS                               *
*********************************************************************
*   APAR#     DATE                                                  *
*  -------  --------                                                *
*********************************************************************
*                                                                   *
*          THE FOLLOWING THREE ENTRIES ARE REQUIRED                 *
*                                                                   *
*********************************************************************
AUTOOPS    IMSMSTR,ID=AUTIMS,
              MSG=(DFS*,AVM005*,AVM006*,IOS071*,DSP*,DXR*,EVI*)
AUTOOPS    IMSWATCH,ID=AUTSURV
AUTOOPS    IMSPPI,ID=AUTIPPI
*********************************************************************
*                                                                   *
*          THERE IS ONE AUTOOPS ENTRY FOR EACH IMS SUBSYTEM DEFINED *
*                                                                   *
*********************************************************************
AUTOOPS    IMSOP01,ID=AUTIMSZ        NON-XRF IMS SUBSYSTEM
AUTOOPS    IMSOP02,ID=AUTIMSA        XRF IMS SUBSYSTEM HSB=1
AUTOOPS    IMSOP03,ID=AUTIMSB        XRF IMS SUBSYSTEM HSB=2
AUTOOPS    IMSOP04,ID=AUTIMSC        DBCTL IMS SUBSYSTEM
*********************************************************************
*          END OF EVIS2OPS                                          *
*********************************************************************
```

EVIS2IMS

```
*********************************************************************
*                                                                   *
*        Licensed Materials - Property of IBM                       *
*        5685-151                                                   *
*        (C) Copyright IBM CORP. 1990, 1996                         *
*                                                                   *
*********************************************************************
*     APAR#      Date   By Description                              *
*  ----------- -------- -- ---------------------------------------- *
*  $L1=OWxxxxx 05/31/96 CG IMS ARM SPE                              *
*                                                                   *
*  $07=PN34670 02/02/94 RW IMSAO CLIST EVIET001 DOES NOT CHECK THAT ALL*
*                          SUBORDINATE ELEMENTS ARE IN AUTODOWN STATUS *
*                          BEFORE ISSUEING USER EXITS.              *
*                                                                   *
*  $06=PN48905 11/23/93 RW STRTTIMER NOT DOCUMENTED OR INCLUDED IN  *
*                          ENVIRON ENTRY OF IMSAO CONTROL FILE.     *
*                                                                   *
*  $05=PN45452 09/09/93 MP CHANGES REQUIRED IN IMSAO FOR COMPATIBILITY *
*                          WITH AOC/MVS V1R2 - DUPLICATED EXTENDED  *
*                          ENTRIES CONTAINING UNDERSCORES, AND CHANGED *
*                          UNDERSCORES TO PERIODS.  ENTRIES CONTAINING *
*                          UNDERSCORES WILL CAUSE CONTROL FILE LOAD *
*                          ERRORS IN AOC/MVS V1R2, AND MUST BE      *
*                          COMMENTED OUT WHEN RUNNING AOC/MVS V1R2. *
*                                                                   *
*  $04=PN36820 08/17/93 MP IMSAO STATE/ACTION TABLE UPDATES AND     *
*                          CORRECTIONS.                             *
*                                                                   *
*  $03=PN43581 08/02/93 RW IMSAO SAMPLE ENTRY FOR SUBSYSTEMS HAVE BOTH *
*                          STARTOPTIONS=PARENT AND STARTCMD=YES CODED  *
*                                                                   *
*  $02=PN39134 04/12/93 JM AFTER OY51838/UY91600 ACORESTART COMMAND *
*                          NEEDS SUBSYS AS A PARM                   *
*                                                                   *
*  $01=PN20372 03/24/93 RW WHEN IMS CONTROL REGION ABENDS WITH ABEND075*
*                          IMSAO SHOULD NOT ATTEMPT TO AUTOMATICALLY *
*                          RESTART                                  *
*                                                                   *
*  PN34417  02/01/93   RW  IMSAO CLIST EVIEU00P DOES NOT HANDLE     *
*                          TASKNAME ON WAIT FOR INPUT REGIONS IN IMS.  *
*                                                                   *
*  PN32675  01/21/93   RW  IMSAO EVIS2IMS SAMPLE MEMBER RE-WORKED FOR  *
*                          CONSISTENCY OF SAMPLES.                  *
*                                                                   *
*  PN29029  01/15/93   RW  IMSAO BMP INITIALIZATION DOES NOT CHECK  *
*                          FIRST FOR PROGRAM / TRANSACTION          *
*                          AVAILABILILTY BEFORE STARTING.           *
*                                                                   *
*  PL82988  10/21/92   RW  IMSAO RESTARTED DEPENDENT REGION DURING  *
*                          MONITOR CYCLE USING MVS START RATHER THAN  *
*                          STARTCMD                                 *
*                                                                   *
*  PN25217  09/24/92   MP  IMSAO STATE/ACTION TABLE PROBLEMS        *
*                                                                   *
*  PN13391  01/17/92   NS  MEMBER EVIS2IMS MISSING PERIOD IN TCO LOAD  *
```

```
*                          EXAMPLE PARAMETER                            *
*                                                                       *
*  PL72497  03/13/91  ADDED EXAMPLES FOR $SI AND $TI                    *
*                                                                       *
*************************************************************************
* STATUS FILE ENTRIES : DURING INITIALIZATION, IMS EXIT WILL READ
*                       AND ALLOCATE, AND START THE DST TASKS
*
*
*
* IMS GROUP NAMES     : GROUP 1 OR MORE IMS'S INTO A GROUP NAME
*
IMSGROUP TSNONXRF,MEMBER=IMS10Z
*
*************************************************************************
*              NON-XRF IMS ENTRIES (IMS10Z)       B E G I N
*************************************************************************
*************************************************************************
* IMS10Z STATEMENTS
*************************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  IMS10Z,JOB=IMS10ZJ,DESC='TEST IMS SUBSYSTEM',
*            ARMNAME=IMS10,         ARM ELEMENT NAME IF APPLICABLE @L1A
             PARENT=VTAM,SHUTDLY=00:20,STARTCMD=YES
*                                                                  @L1A
*ASSOCIATION IMS10Z,                                               @L1A
*            SECONDARY=(SYS2,...,SYSn)                             @L1A
*
IMS10Z     STARTUP,CMD=(,,EVIEI00I)
*
*   IMS V2 AND V3 USE THE FOLLOWING START CMD FORMAT
*
*IMS10Z     START1,CMD=(,,"MVS S IMS10ZJ,AUTO=&EHKVAR1")
*
*   IMS V4 USES THE FOLLOWING START CMD FORMAT
*
*IMS10Z     START1,CMD=(,,"MVS S IMS10ZJ,PARM1='AUTO=&EHKVAR1'")
*
IMSCNTL    IMS10Z,
             APPLID=IMS10Z,
             AUTOOPS=IMSOP01,
             APPLID1=IMS10Z
*
THRESHOLDS IMS10Z,CRIT=(2,08:00),FREQ=(1,04:00),INFR=(01,4:00)
*
IMS10Z     ACORESTART,CMD=(,,'EVIEE00A IMS10Z')            /* @02C*/
*
AUTOMATION IMS10Z,AUTO=Y
INITSTART  IMS10Z,AUTO=E,EXITS=(EVIEIEXT)
START      IMS10Z,AUTO=Y
RECOVERY   IMS10Z,AUTO=Y
TERMINATE  IMS10Z,AUTO=Y
RESTART    IMS10Z,AUTO=E,EXITS=(EVIEIEXT)
*** SUBSYSTEM FLAGS ***
IMS10Z     ENVIRON,
             SHUTGO=YES,
             SUBID=IMSZ,
             XRF=NO,
```

```
             DEFSTART=AUTO,
             SUBTYPE=CTL,
             BTIMER=3,
             ITIMER=6,
             RTIMER=3,
             STRTTIMER=3,
             TTIMER=3
*** SUBSYSTEM MESSAGE AUTOMATION ***
IMS10Z     RESTARTABORT,
                REPLY=(ERE,RETRY9,'/ERE.')
*V2             REPLY=(OVERRIDE,RETRY9,'/ERE OVERRIDE.')
*V3             REPLY=(OVERRIDE,RETRY9,'/ERE COLDSYS OVERRIDE.')
IMS10Z     DFS810A,
             REPLY=(AUTO,RETRY9,'/NRE.'),
             REPLY=(BACKUP,RETRY9,'/ERE BACKUP.'),
             REPLY=(BUILDQ,RETRY9,'/NRE FORMAT RS BUILDQ'),
*V2             REPLY=(COLD,RETRY9,'/NRE CHKPT 0 FORMAT ALL DETACH'),
*V3             REPLY=(COLD,RETRY9,'/NRE CHKPT 0 FORMAT ALL'),
             REPLY=(MANUAL,RETRY9,'&EHKVAR1')
IMS10Z     DFS994I,
             REPLY=(BUILDQ,RETRY9,'/STA DC'),
             REPLY=(BUILDQ,RETRY9,'/START PROGRAM EVISPPII'),
             REPLY=(BUILDQ,RETRY9,'/START TRAN EVITPPI1'),
             REPLY=(BUILDQ,RETRY9,'/CHE'),
             REPLY=(COLD,RETRY9,'/STA DC'),
             REPLY=(COLD,RETRY9,'/START PROGRAM EVISPPII'),
             REPLY=(COLD,RETRY9,'/START TRAN EVITPPI1'),
             REPLY=(COLD,RETRY9,'/CHE'),
             REPLY=(ERE,RETRY9,'/STA DC'),
             REPLY=(ERE,RETRY9,'/START PROGRAM EVISPPII'),
             REPLY=(ERE,RETRY9,'/START TRAN EVITPPI1'),
             REPLY=(ERE,RETRY9,'/ASSIGN LTERM APPLE TO NODE JONE'),
             REPLY=(ERE,RETRY9,'/CHE'),
             REPLY=(MANUAL,RETRY9,'/STA DC'),
             REPLY=(MANUAL,RETRY9,'/START PROGRAM EVISPPII'),
             REPLY=(MANUAL,RETRY9,'/START TRAN EVITPPI1'),
             REPLY=(MANUAL,RETRY9,'/CHE'),
             REPLY=(WARM,RETRY9,'/STA DC'),
             REPLY=(WARM,RETRY9,'/START PROGRAM EVISPPII'),
             REPLY=(WARM,RETRY9,'/START TRAN EVITPPI1'),
             REPLY=(WARM,RETRY9,'/ASSIGN CPRI 8 TO TRAN PIT, SEED'),
             REPLY=(WARM,RETRY9,'/CHE'),
             REPLY=(WARM,RETRY9,'/START TRAN ALL CLASS 6'),
             REPLY=(WARM,RETRY9,'/START DATABASE DTBSE1'),
             REPLY=(WARM,RETRY9,'/START LINE 4 PTERM 1, 2'),
             REPLY=(WARM,RETRY9,'/START LTERM APPLE, ORANG')
*
IMS10Z     PRECHKP,
             REPLY=(,RETRY9,'/PSTOP LTERM APPLE, ORANG'),
             REPLY=(,RETRY9,'/PSTOP LINK ALL')
IMS10Z     POSTCHKP,
             REPLY=(,RETRY9,'/IDLE NODE ABC'),
             REPLY=(,RETRY9,'/IDLE LINK 2')
IMS10Z     STADC,
             REPLY=(,RETRY9,'/STA DC')
IMS10Z     HOLDQ,
             CMD=(,,'MVS $HQ,C=123')
IMS10Z     RELEASEQ,
```

```
             CMD=(,,'MVS $AQ,C=123')
IMS10Z     CHE,
             REPLY=(,RETRY9,'/CHE')
IMS10Z     SNAPQ,
             REPLY=(,RETRY9,'/CHE SNAPQ')
* NEW ENTRIES                                                           @07A*
IMS10Z     STOPREGION,
          REPLY=(NORMAL,RETRY9,'/STOP REG &EHKVAR1'),
          REPLY=(ABEND,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
          REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
          REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 CANCEL')
IMS10Z     STOPFPREGION,
          REPLY=(NORMAL,RETRY9,'/STOP REG &EHKVAR1'),
          REPLY=(ABEND,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
          REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
          REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 CANCEL')
IMS10Z     STOPBMPREGION,
          REPLY=(NORMAL,RETRY9,'/STOP REG &EHKVAR1'),
          REPLY=(ABEND,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
          REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
          REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 CANCEL')
* NEW ENTRIES                                                           @07A*
IMS10Z     VTAMTERMS,
             CMD=(SHUTDOWN,,'V NET,TERM,LU1=&EHKVAR1,TYPE=UNCOND'),
             CMD=(SHUTDOWN,,'V NET,INACT,ID=MTOCONS,FORCE')
*
*   IMS SHUTDOWN COMMANDS
*
IMS10Z     BRO,
             REPLY=(,,'/BRO ACTIVE'),
             REPLY=(,RETRY9,'IMS COMING DOWN IN &EHKVAR1 MINUTES.')
IMS10Z     SHUTNORM,
             CMD=(PASS1,,'EVIET001 &SUBSAPPL,NORM,DUMPQ')
IMS10Z     SHUTIMMED,
             CMD=(PASS1,,'EVIET001 &SUBSAPPL,IMMED,FREEZE')
IMS10Z     SHUTFORCE,
             CMD=(PASS1,,'EVIET001 &SUBSAPPL,FORCE,NODUMP')
IMS10Z     SHUTTYPES,
             REPLY=(DUMPQ,RETRY9,'/CHE DUMPQ'),
             REPLY=(FREEZE,RETRY9,'/CHE FREEZE'),
             REPLY=(PURGE,RETRY9,'/CHE PURGE'),
             CMD=(NODUMP,,'MVS F &EHKVAR1,STOP'),
             CMD=(DUMP,,'MVS F &EHKVAR1,DUMP')
*** SUBSYSTEM ABEND CODES REQUIRED RESTART (NON-XRF OR NO-XRF) ***
IMS10Z     ABCODES,
             CODE=(SYS0C4,,,ABENDING),
             CODE=(IMS0113,,,ABENDING),
             CODE=(IMS0020,,,ABENDING),
             CODE=(IMS0707,,,ABENDING),
             CODE=(IMS0075,,,STOPPING),
             CODE=(*,,,STOPPING)
IMS10Z     TPABEND,
             CODE=(U0002,,,ABENDING)
*
* ENTRIES FOR RECOVERY OF IMS MSC LINKS
*
IMS10Z     DFS2161I,
```

```
             REPLY=(RESTART,,'/RSTART LINK &EHKVAR1')
IMS10Z     DFS2169I,
             REPLY=(RESTART,,'/RSTART LINK &EHKVAR1')
IMS10Z     DFS2142,
             REPLY=(START,,'/START MSNAME &EHKVAR2'),
             REPLY=(RESTART,,'/PSTOP LINK   &EHKVAR1'),
             REPLY=(RESTART,,'/START MSNAME &EHKVAR2'),
             REPLY=(RESTART,,'/RSTART LINK  &EHKVAR1')
*
RECOVERY   IMS10Z.MSC.0001,AUTO=Y
*
* ENTRIES FOR RECOVERY AND ARCHIVE MONITORING OF IMS OLDS
*
IMS10Z     OLDS,
             MINIMUM=03,
             SPARES=(05,99),
             ARCHIVETIME=00:20:00,
             RETRYCNT=5
IMS10Z     DFS3258A,
             CMD=(SYSTEM,,'IMSCMD &SUBSAPPL /LOG WAITING.'),
             CMD=(LAST,,'IMSCMD &SUBSAPPL /STA OLDS 99.')
*
THRESHOLDS IMS10Z.OLDS,
             CRIT=(05,00:30),FREQ=(03,00:30),INFR=(02,00:30)
RECOVERY   IMS10Z.OLDS,AUTO=Y
*
* ENTRIES FOR RECON RECOVERY
*
IMS10Z     RECONS,
             MONITOR=00:15:00
*
* ENTRIES FOR TRANSACTION/PROGRAM RECOVERY
*
IMS10Z     DFS554A,
             CMD=(TRAN,,'IMSCMD &SUBSAPPL /STA TRAN &EHKVAR1'),
             CMD=(PROG,,'IMSCMD &SUBSAPPL /STA PGM &EHKVAR2')
*
RECOVERY   IMS10Z.DFS554A,AUTO=Y
*     CONTROLLING TRANACTION RECOVERY FOR:
*       1) MESSAGE PROCESSING REGIONS
*       2) FASTPATH REGIONS
*       3) TRANSACTION DRIVEN BMP REGIONS
ABCODETRAN IMS10Z.TRAN,
             CODE=(SAMPLE1,U0778,*,EXCLUDE),
             CODE=(*,*,*,INCLUDE)
THRESHOLDS IMS10Z.TRAN,
             CRIT=(09,00:15),FREQ=(06,00:20),INFR=(03,00:35)
THRESHOLDS IMS10Z.TRAN.SAMPLE2,
             CRIT=(05,00:05),FREQ=(04,00:15),INFR=(03,00:25)
THRESHOLDS IMS10Z.TRAN.SAMPLE3,
             CRIT=(05,00:05),FREQ=(04,00:15),INFR=(03,00:25)
RECOVERY   IMS10Z.TRAN,AUTO=Y
RECOVERY   IMS10Z.TRAN.SAMPLE2,AUTO=Y
RECOVERY   IMS10Z.TRAN.SAMPLE3,AUTO=Y
RECOVERY   IMS10Z.TRAN.SAMPLE4,AUTO=N
*     CONTROLLING PROGRAM RECOVERY
ABCODEPROG IMS10Z.PROG,
             CODE=(*,U0778,SAMPLE4,EXCLUDE),
```

```
             CODE=(*,*,*,INCLUDE)
THRESHOLDS IMS10Z.PROG,
             CRIT=(09,00:15),FREQ=(07,00:25),INFR=(04,00:35)
THRESHOLDS IMS10Z.PROG.SAMPLE5,
             CRIT=(05,00:05),FREQ=(04,00:05),INFR=(03,00:05)
RECOVERY   IMS10Z.PROG,AUTO=Y
RECOVERY   IMS10Z.PROG.SAMPLE6,AUTO=Y
*
* ENTRIES FOR USER DEFINED RESTART COMMANDS
*
IMS10Z     USERSTART,
             ERE1='ERE COLDCOMM',
             NRE1='NRE CHKPT 0'
*
* ENTRIES FOR SERVICE PERIODS
*
SERVICE    IMS10Z,
             DAY=(DAILY,0745-1715),
             DAY=(WEEKEND,DOWN-DOWN)
*
* ENTRIES FOR TRIGGERS
*
TRIGGER    IMS10Z,
             STARTUP=(SERVICE,IMSJOB1),
             SHUTDOWN=IMSJOB2
*
* ENTRIES FOR EXTCOND
*
EXTCOND    IMSJOB1,
             UNSET=START,
             DESC='IMS JOB 1 MUST BE COMPLETE'
*
* SAMPLE ENTRIES FOR THE TCO OPERATOR INTERFACE
*    - TCO IS ONLY SUPPORTED ON IMS V3                                 *
*
*V3 IMS10Z    TCO,
*V3             REPLY=(INIT,,'DFSTCF LOAD DFSTCF  .'),
*V3             REPLY=(SPEC,,'DFSTCF LOAD &EHKVAR1 .'),
*V3             REPLY=(START,,'/START LTERM DFSTCFI .'),
*V3             REPLY=(STOP,,'/PSTOP LTERM DFSTCFI .')
***********************************************************************
* DBRC STATEMENTS
***********************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  DBRIMSZ,JOB=DBRIMSZ,DESC='DBRC SUBSYSTEM OF IMS10Z',
             PARENT=IMS10Z,SHUTDLY=00:01,SHUTOPTIONS=PARENT,
             STARTOPTIONS=PARENT
*
THRESHOLDS DBRIMSZ,CRIT=(03,00:10),FREQ=(02,00:10),INFR=(01,00:10)
*
AUTOMATION DBRIMSZ,AUTO=Y
*** SUBSYSTEM FLAGS ***
DBRIMSZ    ENVIRON,
             SUBTYPE=DBRC
***********************************************************************
* DLISAS STATEMENTS
***********************************************************************
```

```
*** SUBSYSTEM NAME ***
SUBSYSTEM  DLIIMSZ,JOB=DLIIMSZ,DESC='DLISAS SUBSYSTEM OF IMS10Z',
             PARENT=IMS10Z,SHUTDLY=00:01,SHUTOPTIONS=PARENT,
             STARTOPTIONS=PARENT
*
THRESHOLDS DLIIMSZ,CRIT=(03,00:10),FREQ=(02,00:10),INFR=(01,00:10)
*
AUTOMATION DLIIMSZ,AUTO=Y
*** SUBSYSTEM FLAGS ***
DLIIMSZ    ENVIRON,
             SUBTYPE=DLS
***********************************************************************
* MSG REGION 1 STATEMENTS
***********************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  MSGIMSZA,JOB=MSGIMSZA,DESC='ONLINE MSG REGION 1',
             PARENT=IMS10Z,SHUTDLY=00:01,                    /* @03C*/
             SHUTOPTIONS=PARENT,
             STARTOPTIONS=PARENT
*** SUBSYSTEM THRESHOLDS ***
THRESHOLDS MSGIMSZA,CRIT=(03,00:10),FREQ=(02,00:10),INFR=(01,00:10)
*** SUBSYSTEM AUTOMATION FLAGS ***
AUTOMATION MSGIMSZA,AUTO=Y
*** SUBSYSTEM FLAGS ***
MSGIMSZA   ENVIRON,
             SUBTYPE=TP
*** SUBSYSTEM START UP PROCESSING ***
MSGIMSZA   START1,CMD=(,,'MVS S IMSRDR,MBR=MSGIMSZA')
*
***********************************************************************
* MSG REGION 2 STATEMENTS
***********************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  MSGIMSZB,JOB=MSGIMSZB,DESC='ONLINE MSG REGION 2',
             PARENT=IMS10Z,SHUTDLY=00:01,STARTCMD=YES
*** SUBSYSTEM THRESHOLDS ***
THRESHOLDS MSGIMSZB,CRIT=(03,00:10),FREQ=(02,00:10),INFR=(01,00:10)
*** SUBSYSTEM AUTOMATION FLAGS ***
AUTOMATION MSGIMSZB,AUTO=Y
*** SUBSYSTEM FLAGS ***
MSGIMSZB   ENVIRON,
             SUBTYPE=TP
*** SUBSYSTEM START UP PROCESSING ***
MSGIMSZB   STARTUP,CMD=(,,'MVS S IMSRDR,MBR=MSGIMSZB')
*
*** SUBSYSTEM SHUTDOWN PROCESSING ***
MSGIMSZB   SHUTINIT,
             CMD=(,,'MSG ALL MSGIMSZB IS COMING DOWN')
*
MSGIMSZB   SHUTNORM,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,NORM')
*
MSGIMSZB   SHUTIMMED,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,IMMED')
*
MSGIMSZB   SHUTFORCE,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,FORCE')
*
```

```
**********************************************************************
* FPM REGION 1 STATEMENTS
**********************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  FPMIMSZA,JOB=FPMIMSZA,DESC='FAST PATH REGION 1',
             PARENT=IMS10Z,SHUTDLY=00:01,STARTCMD=YES
*** SUBSYSTEM THRESHOLDS ***
THRESHOLDS FPMIMSZA,CRIT=(03,00:10),FREQ=(02,00:10),INFR=(01,00:10)
*** SUBSYSTEM AUTOMATION FLAGS ***
AUTOMATION FPMIMSZA,AUTO=Y
*** SUBSYSTEM FLAGS ***
FPMIMSZA   ENVIRON,
             SUBTYPE=FP
*** SUBSYSTEM START UP PROCESSING ***
FPMIMSZA   STARTUP,CMD=(,,'MVS S IMSRDR,MBR=FPMIMSZA')
*
*** SUBSYSTEM SHUTDOWN PROCESSING ***
FPMIMSZA   SHUTINIT,
             CMD=(,,'MSG ALL FPMIMSZA IS COMING DOWN')
*
FPMIMSZA   SHUTNORM,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,NORM')
*
FPMIMSZA   SHUTIMMED,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,IMMED')
*
FPMIMSZA   SHUTFORCE,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,FORCE')
*
**********************************************************************
* BMP REGION 1 STATEMENTS
**********************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  BMPIMSZA,JOB=BMPIMSZA,DESC='BATCH MESSAGE REGION - 1',
             PARENT=IMS10Z,SHUTDLY=00:01,STARTCMD=YES
*** SUBSYSTEM THRESHOLDS ***
THRESHOLDS BMPIMSZA,CRIT=(03,00:10),FREQ=(02,00:10),INFR=(01,00:10)
*** SUBSYSTEM AUTOMATION FLAGS ***
AUTOMATION BMPIMSZA,AUTO=Y
*** SUBSYSTEM FLAGS ***
BMPIMSZA   ENVIRON,
             SUBTYPE=BMP
*** SUBSYSTEM START UP PROCESSING ***
BMPIMSZA   STARTUP,CMD=(,,'MVS S IMSRDR,MBR=BMPIMSZA')
*
*** SUBSYSTEM SHUTDOWN PROCESSING ***
BMPIMSZA   SHUTINIT,
             CMD=(,,'MSG ALL BMPIMSZA IS COMING DOWN')
*
BMPIMSZA   SHUTNORM,
             CMD=(PASS1,,'user - supplied shutdown command')
*
BMPIMSZA   SHUTIMMED,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,IMMED')
*
BMPIMSZA   SHUTFORCE,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,FORCE')
*
```

```
*********************************************************************
* OTHER SUBSYSTEM DEFINITION FOR CHILD OF IMS CONTROL REGION        *
* THIS SUBSYSTEM BELONGS TO IMS CONTROL REGION BUT IS NOT AN        *
* IMS ELEMENT.                                                      *
*********************************************************************
SUBSYSTEM  ABCDE,JOB=ABCDE,DESC='ABCDE SUBSYSTEM CHILD OF IMS',
             PARENT=IMS10Z,STARTCMD=YES,PARMS='USER-PARMS',
             SHUTDLY=00:00:30
*
AUTOMATION ABCDE,AUTO=Y
*
THRESHOLDS  ABCDE,CRIT=(03,00:10),FREQ=(02,00:20),INFR=(01,00:30)
*
ABCDE      ENVIRON,
             SUBTYPE=OTHER
ABCDE      STARTUP,CMD=(,,'STARTUP CMD FOR SUBSYS ABCDE')
ABCDE      SHUTNORM,CMD=(PASS1,,'SHUTDOWN CMD FOR SUBSYS ABCDE')
*
*
*********************************************************************
*              NON-XRF IMS ENTRIES (IMS10Z)     E N D               *
*********************************************************************
```

EVIS2DBC

```
******************************************************************
*                                                                *
*       LICENSED MATERIALS - PROPERTY OF IBM                     *
*       5685-151                                                 *
*       (C) COPYRIGHT IBM CORP. 1990, 1996                       *
*                                                                *
******************************************************************
*    APAR#      Date   By Description                            *
* ----------- -------- -- ---------------------------------------- *
* $04=OW36932,V1R4,06Jan99,APC(RW): Alter REPLY to CMD for BMPIMSC*   *
*                                  SHUT* entries                 *
* $03=OW20827 06/05/96 CG PTF UW90273 DID NOT INCLUDE ++HOLD ACTION   *
*                         STATEMENTS INSTALLER MAY MISS NECESSARY     *
*                         INSTALLATION INFORMATION               *
*                                                                *
* $L1=OW19977 05/31/96 CG AOC IMS-AUTOMATION FEATURE SUPPORT FOR IMS  *
*                         5.1 ARM RECOVERY                       *
*                                                                *
* $02=OW19198 03/13/96 CG MSGDFS101 DURING AUTO=Y IMS DBCTL START UP  *
*                         DFS101                                 *
*                                                                *
* $01=PN34670 02/02/94 RW IMSAO CLIST EVIET001 DOES NOT CHECK THAT ALL*
*                         SUBORDINATE ELEMENTS ARE IN AUTODOWN STATUS *
*                         BEFORE ISSUEING USER EXITS.            *
*                                                                *
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*     IMS01C SUBSYSTEM
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
SUBSYSTEM IMS01C,
   DESC='IMS 4.1 DBCTL',
   PARENT=VTAM,
   STARTCMD=YES,
*  ARMNAME=IMS01,          ARM ELEMENT NAME IF APPLICABLE          @L1A
   SHUTDLY=00:20:00,
   JOB=IMS401C
*                                                              @L1A
*ASSOCIATION IMS01C,                                           @L1A
*           SECONDARY=(SYS2,...,SYSn)                          @L1A
*                                                              @L1A
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*  AUTOMATION AUTOMATION FLAGS FOR IMS01C
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*
AUTOMATION IMS01C,
   AUTO=Y
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*  INITSTART AUTOMATION FLAGS FOR IMS01C
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*
INITSTART IMS01C,
   AUTO=E,
   EXITS=EVIEIEXT
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*  START AUTOMATION FLAGS FOR IMS01C
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
```

```
*
START IMS01C,
   AUTO=Y
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*  RESTART AUTOMATION FLAGS FOR IMS01C
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*
RESTART IMS01C,
   AUTO=E,
   EXITS=EVIEIEXT
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*  RECOVERY AUTOMATION FLAGS FOR IMS01C
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*
RECOVERY IMS01C,
   AUTO=Y
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*  TERMINATE AUTOMATION FLAGS FOR IMS01C
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*
TERMINATE IMS01C,
   AUTO=Y
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*     THRESHOLDS FOR IMS01C SUBSYSTEM
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*
THRESHOLDS IMS01C,
   CRIT=(3,00:30),
   FREQ=(2,00:30),
   INFR=(1,00:30)
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*     MESSAGE TABLE FOR IMS01C
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*
IMS01C ACORESTART,
   CMD=(,,"EVIEE00A IMS01C")
IMS01C STARTUP,
   CMD=(,,"EVIEI00I")
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*   IMS V3 USES THE FOLLOWING START CMD FORMAT
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*IMS01C     START1,
*           CMD=(,,"MVS S IMS401C,AUTO=&EHKVAR1")
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*   IMS V4 USES THE FOLLOWING START CMD FORMAT
* - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - *
*IMS01C     START1,
*           CMD=(,,"MVS S IMS401C,PARM1='AUTO=&EHKVAR1'")
*
IMSGROUP DBCTL41,MEMBER=IMS01C
*
*********************************************************************
* IMS01C DBCTL DEFINITIONS FOR IMS01C
*********************************************************************
*
IMSCNTL    IMS01C,
```

```
             AUTOOPS=IMSOP04
*
*
*** SUBSYSTEM FLAGS ***
IMS01C     ENVIRON,
              XRF=NO,
              EXITS=(CHKSTOP),
              DBCTL=YES,
              SHUTGO=YES,
              DEFSTART=AUTO,
              SUBID=I41C,
              SUBTYPE=CTL,
              BTIMER=6,
              ITIMER=8,
              RTIMER=3,
              TTIMER=3,
              STRTTIMER=3
************************************************************************
*   SUBSYSTEM MESSAGE AUTOMATION
************************************************************************
IMS01C     RESTARTABORT,
             CMD=(ERE,,'MVS &EHKVAR7ERE.'),
             CMD=(OVERRIDE,,'MVS &EHKVAR7ERE COLDSYS OVERRIDE.')
*    USE DFS989I INSTEAD OF DFS810A FOR DBCTL
*            CMD=(AUTO,,'MVS &EHKVAR7NRE.'),                    @02M*
* START AUTO COMMAND NOT REQUIRED.                              @02A*
* IMS INTERNALLY UNDERSTANDS HOW TO PROCEED.                    @02A*
IMS01C     DFS989I,
             CMD=(COLD,,'MVS &EHKVAR7NRE CHKPT 0 FORMAT ALL'),
             CMD=(MANUAL,,'&EHKVAR1')
IMS01C     DFS994I,
             CMD=(COLD,,'MVS &EHKVAR7CHE'),
             CMD=(ERE,,'MVS &EHKVAR7CHE'),
             CMD=(MANUAL,,'MVS &EHKVAR7CHE'),
             CMD=(WARM,,'MVS &EHKVAR7CHE')
*
IMS01C     CHE,
             CMD=(,,'MVS &EHKVAR7CHE')
*
* NEW ENTRIES                                                   @01A*
IMS10C     STOPBMPREGION,
          CMD=(NORMAL,,'MVS &EHKVAR7 REG &EHKVAR1'),
          CMD=(ABEND,,'MVS &EHKVAR7 REG &EHKVAR1 ABDUMP &EHKVAR2'),
          CMD=(CANCEL,,'MVS &EHKVAR7 REG &EHKVAR1 ABDUMP &EHKVAR2'),
          CMD=(CANCEL,,'MVS &EHKVAR7 REG &EHKVAR1 CANCEL')
* NEW ENTRIES                                                   @01A*
*
*   IMS SHUTDOWN COMMANDS
*
IMS01C     SHUTNORM,
             CMD=(PASS1,,'EVIET001 &SUBSAPPL,NORM,FREEZE')
IMS01C     SHUTIMMED,
             CMD=(PASS1,,'EVIET001 &SUBSAPPL,IMMED,PURGE')
IMS01C     SHUTFORCE,
             CMD=(PASS1,,'EVIET001 &SUBSAPPL,FORCE,DUMP')
IMS01C     SHUTTYPES,
             CMD=(FREEZE,,'MVS &EHKVAR7CHE FREEZE'),
             CMD=(PURGE,,'MVS &EHKVAR7CHE PURGE'),
```

```
              CMD=(NODUMP,,'MVS F &EHKVAR1,STOP'),
              CMD=(DUMP,,'MVS F &EHKVAR1,DUMP')
*** SUBSYSTEM ABEND CODES REQUIRED RESTART (NON-XRF OR NO-XRF) ***
IMS01C      ABCODES,
              CODE=(SYS0C4,,,ABENDING),
              CODE=(SYS222,,,ABENDING),
              CODE=(SYS122,,,ABENDING),
              CODE=(IMS0113,,,ABENDING),
              CODE=(IMS0020,,,ABENDING),
              CODE=(IMS0707,,,ABENDING),
              CODE=(*,,,STOPPING)
IMS01C      TPABEND,
              CODE=(U0002,,,ABENDING)
*
* SERVICE PERIOD FOR IMS01C
SERVICE     IMS01C,DAY=(DAILY,1000-2300)
*
*
*
* ENTRIES FOR RECOVERY OF IMS OLDS
*
IMS01C      DFS3258A,
              CMD=(SYSTEM,,'MVS &EHKVAR7LOG WAITING.'),
              CMD=(LAST,,'MVS &EHKVAR7STA OLDS 99.')
IMS01C      OLDS,
              MINIMUM=03,
              SPARES=(05,99),
              ARCHIVETIME=00:20:00,
              RETRYCNT=5
THRESHOLDS IMS01C.OLDS,
              CRIT=(05,00:30),FREQ=(03,00:30),INFR=(02,00:30)
RECOVERY    IMS10C.OLDS,AUTO=Y
*
* ENTRIES FOR RECON RECOVERY
*
IMS01C      RECONS,
              MONITOR=00:15:00
*
* ENTRIES FOR PROGRAM ABEND RECOVERY
*
IMS01C      DFS554A,
              CMD=(PROG,,'MVS &EHKVAR7STA PGM &EHKVAR2')
*     CONTROLLING PROGRAM RECOVERY
ABCODEPROG IMS01C.PROG,
              CODE=(*,U0778,SAMPLE4,EXCLUDE),
              CODE=(*,*,*,INCLUDE)
THRESHOLDS IMS01C.PROG,
              CRIT=(09,00:15),FREQ=(07,00:25),INFR=(04,00:35)
THRESHOLDS IMS01C.PROG.SAMPLE5,
              CRIT=(05,00:05),FREQ=(04,00:05),INFR=(03,00:05)
RECOVERY    IMS01C.PROG,AUTO=Y
RECOVERY    IMS01C.PROG.SAMPLE6,AUTO=Y
*
* ENTRIES FOR USER DEFINED RESTART COMMANDS
*
IMS01C      USERSTART,
              ERE1='ERE COLDBASE',
```

```
             NRE1='NRE CHKPT 0'
*
* ENTRIES FOR SERVICE PERIODS
*
SERVICE    IMS01C,
           DAY=(DAILY,1000-1400)
*
* ENTRIES FOR TRIGGERS
*
* TRIGGER  IMS01C,
*            STARTUP=(SERVICE,IMSCDONE),
*            STARTUP=IMSCDONE,
*            SHUTDOWN=DOWNC,
*            SHUTDOWN=(TEST1C,TEST2C,TEST3C,TEST4C)
*
* ENTRIES FOR EXTCOND
*
EXTCOND    IMSCDONE,
             UNSET=START,
             DESC='IMS JOB C MUST BE COMPLETE'
***********************************************************************
* DBRC STATEMENTS
***********************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  DBRIMSC,JOB=DBRIMSC,DESC='DBRC SUBSYSTEM OF IMS01C',
             PARENT=IMS01C,SHUTDLY=00:01,SHUTOPTIONS=PARENT,
             STARTOPTIONS=PARENT
*
THRESHOLDS DBRIMSC,CRIT=(03,00:10),FREQ=(02,00:10),INFR=(01,00:10)
*
AUTOMATION DBRIMSC,AUTO=Y
*** SUBSYSTEM FLAGS ***
DBRIMSC    ENVIRON,
             SUBTYPE=DBRC
***********************************************************************
* DLISAS STATEMENTS
***********************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  DLIIMSC,JOB=DLIIMSC,DESC='DLISAS SUBSYSTEM OF IMS01C',
             PARENT=IMS01C,SHUTDLY=00:01,SHUTOPTIONS=PARENT,
             STARTOPTIONS=PARENT
*
THRESHOLDS DLIIMSC,CRIT=(03,00:10),FREQ=(02,00:10),INFR=(01,00:10)
*
AUTOMATION DLIIMSC,AUTO=Y
*** SUBSYSTEM FLAGS ***
DLIIMSC    ENVIRON,
             SUBTYPE=DLS
***********************************************************************
* BMP REGION 1 STATEMENTS
***********************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  BMPIMSCA,JOB=BMPIMSCA,DESC='BMP MSG REGION 1',
             PARENT=IMS01C,SHUTDLY=00:02,STARTCMD=YES
*** SUBSYSTEM THRESHOLDS ***
THRESHOLDS BMPIMSCA,CRIT=(03,00:10),FREQ=(02,00:10),INFR=(01,00:10)
*** SUBSYSTEM AUTOMATION FLAGS ***
AUTOMATION BMPIMSCA,AUTO=Y
```

```
*** SUBSYSTEM FLAGS ***
BMPIMSCA   ENVIRON,
             SUBTYPE=BMP
*** SUBSYSTEM START UP PROCESSING ***
BMPIMSCA   STARTUP,
             CMD=(,,'MVS S IMSRDRC,MBR=BMPIMSCA')
*
*** SUBSYSTEM SHUTDOWN PROCESSING ***
BMPIMSCA   SHUTINIT,
              CMD=(,,'MSG ALL BMPIMSCA IS COMMING DOWN')
BMPIMSCA   SHUTNORM,
              CMD=(PASS1,,'EVIET00J &SUBSAPPL,NORM')
BMPIMSCA   SHUTIMMED,
              CMD=(PASS1,,'EVIET00J &SUBSAPPL,IMMED')
BMPIMSCA   SHUTFORCE,
              CMD=(PASS1,,'EVIET00J &SUBSAPPL,FORCE')
*
*
*********************************************************************
* BMP REGION 2 STATEMENTS
*********************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  BMPIMSCB,JOB=BMPIMSCB,DESC='BMP MSG REGION 2',
             PARENT=IMS01C,SHUTDLY=00:02,STARTCMD=YES
*** SUBSYSTEM THRESHOLDS ***
THRESHOLDS BMPIMSCB,CRIT=(03,00:10),FREQ=(02,00:10),INFR=(01,00:10)
*** SUBSYSTEM AUTOMATION FLAGS ***
AUTOMATION BMPIMSCB,AUTO=Y
*** SUBSYSTEM FLAGS ***
BMPIMSCB   ENVIRON,
             SUBTYPE=BMP
*** SUBSYSTEM START UP PROCESSING ***
BMPIMSCB   STARTUP,
             CMD=(,,'MVS S IMSRDRC,MBR=BMPIMSCB')
*
*** SUBSYSTEM SHUTDOWN PROCESSING ***
BMPIMSCB   SHUTINIT,
              CMD=(,,'MSG ALL BMPIMSCB IS COMMING DOWN')
BMPIMSCB   SHUTNORM,
              CMD=(PASS1,,'EVIET00J &SUBSAPPL,NORM')
BMPIMSCB   SHUTIMMED,
              CMD=(PASS1,,'EVIET00J &SUBSAPPL,IMMED')
BMPIMSCB   SHUTFORCE,
              CMD=(PASS1,,'EVIET00J &SUBSAPPL,FORCE')
*
*********************************************************************
* BMP REGION 3 STATEMENTS
*********************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  BMPIMSCC,JOB=BMPIMSCC,DESC='BMP MSG REGION 3',
             PARENT=IMS01C,SHUTDLY=00:02,STARTCMD=YES
*** SUBSYSTEM THRESHOLDS ***
THRESHOLDS BMPIMSCC,CRIT=(03,00:10),FREQ=(02,00:10),INFR=(01,00:10)
*** SUBSYSTEM AUTOMATION FLAGS ***
AUTOMATION BMPIMSCC,AUTO=Y
*** SUBSYSTEM FLAGS ***
BMPIMSCC   ENVIRON,
```

```
             SUBTYPE=BMP
*** SUBSYSTEM START UP PROCESSING ***
BMPIMSCC   STARTUP,
             CMD=(,,'MVS S IMSRDRC,MBR=BMPIMSCC')
*
*** SUBSYSTEM SHUTDOWN PROCESSING ***
BMPIMSCC   SHUTINIT,
              CMD=(,,'MSG ALL BMPIMSCC IS COMMING DOWN')
BMPIMSCC   SHUTNORM,
              CMD=(PASS1,,'EVIET00J &SUBSAPPL,NORM')
BMPIMSCC   SHUTIMMED,
              CMD=(PASS1,,'EVIET00J &SUBSAPPL,IMMED')
BMPIMSCC   SHUTFORCE,
              CMD=(PASS1,,'EVIET00J &SUBSAPPL,FORCE')
***********************************************************************
*     END OF DEFINITION
***********************************************************************
```

EVIS2XRF

```
***********************************************************************
*                                                                     *
*        Licensed Materials - Property of IBM                         *
*        5685-151                                                     *
*        (C) Copyright IBM CORP. 1990, 1996                           *
*                                                                     *
***********************************************************************
*     APAR#      Date   By Description                                *
*  ----------- -------- -- ----------------------------------------- *
*  $L1=IMSSPE  05/31/96 CG IMS ARM SPE                                *
*                                                                     *
*  $03=OW17107 11/30/95 CG EVIET008 NOT TRAPPING MSGDFS2717 NOR       *
*                          MSGDFS058I STOP COMMAND IN PROGRESS        *
*                                                                     *
*  $02=OW06963 07/26/94 JG SAMPLES IN AEVISAMP ARE INCORRECT FOR IMS   *
*                          XRF AND LCL MEMBERS.                       *
*                                                                     *
*  $01=PN34670 02/02/94 RW IMSAO CLIST EVIET001 DOES NOT CHECK THAT ALL*
*                          SUBORDINATE ELEMENTS ARE IN AUTODOWN STATUS *
*                          BEFORE ISSUEING USER EXITS.                *
*                                                                     *
***********************************************************************
***********************************************************************
*              XRF IMS ENTRIES (IMS10AA - IMS10BB)  B E G I N         *
***********************************************************************
*  THE FOLLOWING ENTRY REQUIRED TO DEFINE PARTNER STATUS FILE         *
*  IN AND XRF COMPLEX FOR AOC/IMS XRF RECOVERY.  SEE CHAPTER 9, STEP 14*
*  OF "AOC/MVS IMS PROGRAMMER'S REFERENCE AND INSTALLATION GUIDE",    *
*  (SC23-3817).                                                       *
***********************************************************************
***********************************************************************
STSFILE  ACO10,DSN=ACOV.SA10.ACO.STATS
***********************************************************************
*                              IMS10AA       B E G I N                *
***********************************************************************
IMSRSENM   IACO,SUBSYSTEM1=IMS10AA,SUBSYSTEM2=IMS10AB
***********************************************************************
* IMS10AA STATEMENTS
***********************************************************************
SUBSYSTEM  IMS10AA,JOB=IMS10AA,DESC='TEST IMS SUBSYSTEM',
*            ARMNAME=IMS10,                                       @L1A
             PARENT=VTAM,SHUTDLY=00:20,STARTCMD=YES
*                                                                 @L1A
*ASSOCIATION IMS10AA,                                             @L1A
*            SECONDARY=(SYS2,...,SYSn)                            @L1A
*
IMS10AA    STARTUP,CMD=(,,EVIEI00I)
*
*   IMS V2 AND V3 USE THE FOLLOWING START CMD FORMAT
*
*IMS10AA    START1,
*           CMD=(,,"MVS S IMS10AA,AUTO=&EHKVAR1,HSBID=&EHKVAR2")
*IMS10AA    START2,
*           CMD=(,,"MVS S IMS10AA,AUTO=&EHKVAR1,HSBID=&EHKVAR2")
*
*   IMS V4 USES THE FOLLOWING START CMD FORMAT
```

```
*
*IMS10AA    START1,
*      CMD=(,,"MVS S IMS10AA,PARM1='AUTO=&EHKVAR1,HSBID=&EHKVAR2'")
*IMS10AA    START2,
*      CMD=(,,"MVS S IMS10AA,PARM1='AUTO=&EHKVAR1,HSBID=&EHKVAR2'")
*
IMS10AA    ACORESTART,CMD=(,,'EVIEE00A IMS10AA')
IMSCNTL    IMS10AA,
             APPLID=(IMSXRF,IMS10AA),
             AUTOOPS=IMSOP02,
             APPLID1=IMS10AA,APPLID2=IMS10AA,
             MAJNODE1=,MAJNODE2=,
             OTHMAJ1=,OTHMAJ2=,
             PARMS1=,PARMS2=,
             DEFLTHSB=1
*
THRESHOLDS IMS10AA,CRIT=(09,00:30),FREQ=(07,06:00),INFR=(02,12:00)
*
AUTOMATION IMS10AA,AUTO=Y
INITSTART  IMS10AA,AUTO=E,EXITS=(EVIEIEXT)
START      IMS10AA,AUTO=Y
RECOVERY   IMS10AA,AUTO=Y
TERMINATE  IMS10AA,AUTO=Y
RESTART    IMS10AA,AUTO=E,EXITS=(EVIEIEXT)
IMS10AA    ENVIRON,
              SHUTGO=YES,
              ALTSTART=YES,
              SYNCH=Y,
              TKO=YES,
              ALTRST=NO,
              APPLVER=NO,
              DEFSTART=AUTO,
              DEFXRFST=XRF,
              PRTNRDOM=AC010,
              PRTNRID=IMSB,
              PRTNRSUB=IMS10AB,
              SUBID=IMSA,
              SHUTACT=NO,
              XRF=YES,
              SUBTYPE=CTL,
              WDELAY=20,
              SDELAY=50,
              BTIMER=3,
              ITIMER=8,
              RTIMER=3,
              TTIMER=3,
              STRTTIMER=3,
              WTCHNOTF=120,
              WTCHALERT=240
*** SUBSYSTEM MESSAGE AUTOMATION ***
IMS10AA    UNLKAVM,
             CMD=(,,'MVS R &EHKVAR1,UNLOCK')
IMS10AA    UNLOCK,
             REPLY=(,REPLY5,'/UNLOCK SYSTEM')
IMS10AA    RESTARTABORT,
                REPLY=(ERE,RETRY9,'/ERE.')                   /* @02C*/
*V2             REPLY=(OVERRIDE,RETRY9,'/ERE OVERRIDE.')
```

```
*V3            REPLY=(OVERRIDE,RETRY9,'/ERE COLDSYS OVERRIDE.')
IMS10AA    DFS3869A,
             REPLY=(,RETRY9,'/SWITCH SYSTEM FORCE')
IMS10AA    DFS3869I,
             REPLY=(,RETRY9,'/SWITCH SYSTEM FORCE')
IMS10AA    DFS810A,
             REPLY=(BACKUP,RETRY9,'/ERE BACKUP.'),
             REPLY=(BUILDQ,RETRY9,'/NRE FORMAT RS BUILDQ'),
*V2            REPLY=(COLD,RETRY9,'/NRE CHKPT 0 FORMAT ALL DETACH'),
*V3            REPLY=(COLD,RETRY9,'/NRE CHKPT 0 FORMAT ALL'),
             REPLY=(MANUAL,RETRY9,'&EHKVAR1')
IMS10AA    DFS994I,REPLY=(BUILDQ,RETRY9,'/STA DC'),
             REPLY=(BUILDQ,RETRY9,'/START PROGRAM EVISPPII'),
             REPLY=(BUILDQ,RETRY9,'/START TRAN EVITPPI1'),
             REPLY=(BUILDQ,RETRY9,'/CHE'),
             REPLY=(COLD,RETRY9,'/STA DC'),
             REPLY=(COLD,RETRY9,'/START PROGRAM EVISPPII'),
             REPLY=(COLD,RETRY9,'/START TRAN EVITPPI1'),
             REPLY=(COLD,RETRY9,'/CHE'),
             REPLY=(ERE,RETRY9,'/STA DC'),
             REPLY=(ERE,RETRY9,'/START PROGRAM EVISPPII'),
             REPLY=(ERE,RETRY9,'/START TRAN EVITPPI1'),
             REPLY=(ERE,RETRY9,'/ASSIGN LTERM APPLE TO NODE JONE'),
             REPLY=(ERE,RETRY9,'/CHE'),
             REPLY=(MANUAL,RETRY9,'/STA DC'),
             REPLY=(MANUAL,RETRY9,'/START PROGRAM EVISPPII'),
             REPLY=(MANUAL,RETRY9,'/START TRAN EVITPPI1'),
             REPLY=(MANUAL,RETRY9,'/CHE'),
             REPLY=(WARM,RETRY9,'/STA DC'),
             REPLY=(WARM,RETRY9,'/START PROGRAM EVISPPII'),
             REPLY=(WARM,RETRY9,'/START TRAN EVITPPI1'),
             REPLY=(WARM,RETRY9,'/ASSIGN LTERM APPLE TO NODE JONE'),
             REPLY=(WARM,RETRY9,'/CHE'),
             REPLY=(WARM,RETRY9,'/START TRAN ALL CLASS 6'),
             REPLY=(WARM,RETRY9,'/START DATABASE DTBSE1'),
             REPLY=(WARM,RETRY9,'/START LINE 4 PTERM 1, 2'),
             REPLY=(WARM,RETRY9,'/START LTERM APPLE, ORANG')
*
IMS10AA    PRECHKP,
             REPLY=(,RETRY9,'/PSTOP LINK ALL')
IMS10AA    POSTCHKP,
             REPLY=(,RETRY3,'/IDLE LINK ALL')
IMS10AA    STADC,
             REPLY=(,RETRY9,'/STA DC')
IMS10AA    HOLDQ,
             CMD=(,,'MVS $HQ,C=123')
*IMS10AA    RELEASEQ,
*             CMD=(,,'MVS $AQ,C=123')
*IMS10AA    $SI,
*             CMD=(,,'MVS $SI1-12')
*IMS10AA    $TI,
*             CMD=(,,'MVS $TI1-12,A')
IMS10AA    CHE,
             REPLY=(,RETRY9,'/CHE')
IMS10AA    SNAPQ,
             REPLY=(,RETRY9,'/CHE SNAPQ')
* NEW ENTRIES                                                     @01A*
IMS10AA    STOPREGION,
```

```
          REPLY=(NORMAL,RETRY9,'/STOP REG &EHKVAR1'),
          REPLY=(ABEND,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
          REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
          REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 CANCEL')
IMS10AA    STOPFPREGION,
          REPLY=(NORMAL,RETRY9,'/STOP REG &EHKVAR1'),
          REPLY=(ABEND,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
          REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
          REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 CANCEL')
IMS10AA    STOPBMPREGION,
          REPLY=(NORMAL,RETRY9,'/STOP REG &EHKVAR1'),
          REPLY=(ABEND,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
          REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
          REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 CANCEL')
* NEW ENTRIES                                                   @01A*
IMS10AA    VTAMTERMS,
             CMD=(SHUTDOWN,,'V NET,TERM,LU1=&EHKVAR1,TYPE=UNCOND'),
             CMD=(SHUTDOWN,,'V NET,ID=MTOCONSOLE,INACT'),
             CMD=(TAKEOVER,,'V NET,TERM,LU1=&EHKVAR1,TYPE=FORCE'),
             CMD=(TAKEOVER,,'V NET,INACT,ID=MTOCONSOLE,FORCE')
IMS10AA    FORCE,
             CMD=(,,'MVS F &EHKVAR1,STOP')
*
*    IMS SHUTDOWN COMMANDS
*
IMS10AA    BRO,
             REPLY=(,,'/BRO ACTIVE'),
             REPLY=(,RETRY9,'IMS COMING DOWN IN &EHKVAR1 MINUTES.')
IMS10AA    SHUTNORM,
             CMD=(PASS1,,'EVIET001 &SUBSAPPL,NORM,DUMPQ')
IMS10AA    SHUTIMMED,
             CMD=(PASS1,,'EVIET001 &SUBSAPPL,IMMED,FREEZE')
IMS10AA    SHUTFORCE,
             CMD=(PASS1,,'EVIET001 &SUBSAPPL,FORCE,NODUMP')
IMS10AA    SHUTTYPES,
             REPLY=(BACKUP,RETRY9,'/STOP BACKUP'),
             REPLY=(DUMPQ,RETRY9,'/CHE DUMPQ'),
             REPLY=(FREEZE,RETRY9,'/CHE FREEZE'),
             REPLY=(PURGE,RETRY9,'/CHE PURGE'),
             CMD=(NODUMP,,'MVS F &EHKVAR1,STOP'),
             CMD=(DUMP,,'MVS F &EHKVAR1,DUMP')
*** SUBSYSTEM ABEND CODES REQUIRED RESTART (NON-XRF OR NO-XRF) ***
IMS10AA    ABCODES,CODE=(SYS0C4,,,ABENDING),
             CODE=(IMS0113,,,ABENDING),
             CODE=(IMS0020,,,ABENDING),
             CODE=(IMS0707,,,ABENDING),
             CODE=(IMS0075,,,STOPPING)
IMS10AA    ACTCODES,CODE=(SYS0C4,,,ABENDING),
             CODE=(IMS0113,,,ABENDING),
             CODE=(IMS0020,,,ABENDING),
             CODE=(IMS0707,,,ABENDING),
             CODE=(IMS0604,,,ABENDING),
             CODE=(IMS0075,,,STOPPING)
IMS10AA    ALTCODES,CODE=(SYS0C4,,,ABENDING),
             CODE=(IMS0113,,,ABENDING),
             CODE=(IMS0020,,,ABENDING),
             CODE=(IMS0707,,,ABENDING),
```

```
             CODE=(IMS0075,,,STOPPING)
IMS10AA    TPABEND,
             CODE=(U0002,,,ABENDING)
*
* ENTRIES FOR RECOVERY OF IMS MSC LINKS
*
IMS10AA    DFS2161I,
             REPLY=(RESTART,,'/RSTART LINK &EHKVAR1')
IMS10AA    DFS2169I,
             REPLY=(RESTART,,'/RSTART LINK &EHKVAR1')
IMS10AA    DFS2142,
             REPLY=(START,,'/START MSNAME &EHKVAR2'),
             REPLY=(RESTART,,'/PSTOP LINK   &EHKVAR1'),
             REPLY=(RESTART,,'/START MSNAME &EHKVAR2'),
             REPLY=(RESTART,,'/RSTART LINK  &EHKVAR1')
*
RECOVERY   IMS10AA.MSC.0001,AUTO=N
*
* ENTRIES FOR RECOVERY AND ARCHIVE MONITORING OF OLDS
*
IMS10AA    OLDS,
             MINIMUM=03,
             SPARES=(05,99),
             ARCHIVETIME=00:20:00,
             RETRYCNT=5
IMS10AA    DFS3258A,
             CMD=(SYSTEM,,'IMSCMD &SUBSAPPL /LOG WAITING.'),
             CMD=(LAST,,'IMSCMD &SUBSAPPL /STA OLDS 99.')
*
THRESHOLDS IMS10AA.OLDS,
             CRIT=(05,00:30),FREQ=(03,00:30),INFR=(02,00:30)
RECOVERY   IMS10AA.OLDS,AUTO=Y
*
* ENTRIES FOR RECON RECOVERY
*
IMS10AA    RECONS,
             MONITOR=00:15:00
*
* ENTRIES FOR TRANSACTION/PROGRAM RECOVERY
*
IMS10AA    DFS554A,
             CMD=(TRAN,,'IMSCMD &SUBSAPPL /STA TRAN &EHKVAR1'),
             CMD=(PROG,,'IMSCMD &SUBSAPPL /STA PGM &EHKVAR2')
*
RECOVERY   IMS10AA.DFS554A,AUTO=Y
*
*     CONTROLLING TRANACTION RECOVERY FOR:
*       1) MESSAGE PROCESSING REGIONS
*       2) FASTPATH REGIONS
*       3) TRANSACTION DRIVEN BMP REGIONS
ABCODETRAN IMS10AA.TRAN,
             CODE=(SAMPLE1,U0778,*,EXCLUDE),
             CODE=(*,*,*,INCLUDE)
THRESHOLDS IMS10AA.TRAN,
             CRIT=(09,00:15),FREQ=(08,00:15),INFR=(04,00:15)
THRESHOLDS IMS10AA.TRAN.SAMPLE2,
             CRIT=(05,00:05),FREQ=(04,00:05),INFR=(03,00:05)
RECOVERY   IMS10AA.TRAN,AUTO=Y
```

```
RECOVERY   IMS10AA.TRAN.SAMPLE3,AUTO=Y
RECOVERY   IMS10AA.TRAN.SAMPLE4,AUTO=N
*     CONTROLLING PROGRAM RECOVERY
ABCODEPROG IMS10AA.PROG,
             CODE=(*,U0778,SAMPLE4,EXCLUDE),
             CODE=(*,*,*,INCLUDE)
THRESHOLDS IMS10AA.PROG,
             CRIT=(09,00:15),FREQ=(08,00:25),INFR=(04,00:35)
THRESHOLDS IMS10AA.PROG.SAMPLE5,
             CRIT=(05,00:05),FREQ=(04,00:05),INFR=(03,00:05)
RECOVERY   IMS10AA.PROG,AUTO=Y
RECOVERY   IMS10AA.PROG.SAMPLE6,AUTO=Y
*
* ENTRIES FOR USER DEFINED RESTART COMMANDS
*
IMS10AA    USERSTART,
             ERE1='ERE COLDCOMM',
             NRE1='NRE CHKPT 0'
*
* ENTRIES FOR SERVICE PERIODS
*
SERVICE    IMS10AA,
             DAY=(DAILY,0745-1715),
             DAY=(WEEKEND,DOWN-DOWN)
*
* ENTRIES FOR TRIGGERS
*
TRIGGER    IMS10AA,
             STARTUP=(SERVICE,JOB1DONE),
             SHUTDOWN=DOWN1
*
* ENTRIES FOR EXTCOND
*
EXTCOND    JOB1DONE,
             UNSET=START,
             DESC='IMS JOB 1 MUST BE COMPLETE'
*
*
* SAMPLE ENTRIES FOR THE TCO  OPERATOR INTERFACE
*    - TCO IS ONLY SUPPORTED ON IMS V3 OR HIGHER                @03C*
*
*V3 IMS10AA   TCO,
*V3             REPLY=(INIT,,'DFSTCF LOAD DFSTCF  .'),
*V3             REPLY=(SPEC,,'DFSTCF LOAD &EHKVAR1 .'),         @03C*
*V3             REPLY=(START,,'/START LTERM DFSTCFI .'),
*V3             REPLY=(STOP,,'/PSTOP LTERM DFSTCFI .')
*************************************************************************
* DBRC STATEMENTS
*************************************************************************
SUBSYSTEM  DBRIMSAA,JOB=DBRIMSAA,DESC='DBRC SUBSYSTEM OF IMS10AA',
             PARENT=IMS10AA,SHUTDLY=00:01,SHUTOPTIONS=PARENT,
             STARTOPTIONS=PARENT
*
THRESHOLDS DBRIMSAA,CRIT=(03,00:10),FREQ=(02,00:10),INFR=(01,00:10)
*
AUTOMATION DBRIMSAA,AUTO=Y
DBRIMSAA   ENVIRON,
```

```
             SUBTYPE=DBRC
************************************************************************
* DLISAS STATEMENTS
************************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  DLIIMSAA,JOB=DLIIMSAA,DESC='DLISAS SUBSYSTEM OF IMS10AA',
             PARENT=IMS10AA,SHUTDLY=00:01,SHUTOPTIONS=PARENT,
             STARTOPTIONS=PARENT
*
THRESHOLDS DLIIMSAA,CRIT=(02,08:00),FREQ=(01,04:00),INFR=(01,04:00)
*
AUTOMATION DLIIMSAA,AUTO=Y
*** SUBSYSTEM FLAGS ***
DLIIMSAA   ENVIRON,
             SUBTYPE=DLS
************************************************************************
* MSG REGION 1 STATEMENTS
************************************************************************
SUBSYSTEM  MSGIMSAA,JOB=MSGIMSAA,DESC='ONLINE MSG REGION 1',
             PARENT=IMS10AA,SHUTDLY=00:01,
             SHUTOPTIONS=PARENT,
             STARTOPTIONS=PARENT
THRESHOLDS MSGIMSAA,CRIT=(03,00:10),FREQ=(02,00:10),INFR=(01,00:10)
AUTOMATION MSGIMSAA,AUTO=Y
MSGIMSAA   ENVIRON,
             SYNCH=Y,
             SUBTYPE=TP,
             PRTNRSUB=MSGIMSBA
MSGIMSAA   START1,
             CMD=(,,'MVS S IMSRDR,MBR=MSGIMSAA')
MSGIMSAA   START2,
             CMD=(,,'MVS S IMSRDR,MBR=MSGIMSAA')
************************************************************************
SUBSYSTEM  MSGIMSAB,JOB=MSGIMSAB,DESC='ONLINE MSG REGION 2',
             PARENT=IMS10AA,SHUTDLY=00:01,STARTCMD=YES
THRESHOLDS MSGIMSAB,CRIT=(03,00:10),FREQ=(02,00:10),INFR=(01,00:10)
AUTOMATION MSGIMSAB,AUTO=Y
MSGIMSAB   ENVIRON,
             SYNCH=Y,
             SUBTYPE=TP,
             PRTNRSUB=MSGIMSBB
MSGIMSAB   STARTUP,
             CMD=(,,'MVS S IMSRDR,MBR=MSGIMSAB')
MSGIMSAB   START1,
             CMD=(,,'MVS S IMSRDR,MBR=MSGIMSAB')
MSGIMSAB   START2,
             CMD=(,,'MVS S IMSRDR,MBR=MSGIMSAB')
***  SUBSYSTEM SHUTDOWN PROCESSING                                   ***
MSGIMSAB   SHUTINIT,
             CMD=(,,'MSG ALL MSGIMSAB IS COMING DOWN')
MSGIMSAB   SHUTNORM,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,NORM')
MSGIMSAB   SHUTIMMED,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,IMMED')
MSGIMSAB   SHUTFORCE,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,FORCE')
************************************************************************
* FPM REGION 1 STATEMENTS
```

```
************************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  FPMIMSAA,JOB=FPMIMSAA,DESC='FAST PATH REGION 1',
             PARENT=IMS10AA,SHUTDLY=00:01,STARTCMD=YES
*** SUBSYSTEM THRESHOLDS ***
THRESHOLDS FPMIMSAA,CRIT=(03,00:10),FREQ=(02,00:10),INFR=(01,00:10)
*** SUBSYSTEM AUTOMATION FLAGS ***
AUTOMATION FPMIMSAA,AUTO=Y
*** SUBSYSTEM FLAGS ***
FPMIMSAA   ENVIRON,
             SYNCH=Y,
             PRTNRSUB=FPMIMSBA,
             SUBTYPE=FP
*** SUBSYSTEM START UP PROCESSING ***
FPMIMSAA   STARTUP,CMD=(,,'MVS S IMSRDR,MBR=FPMIMSAA')
*
*** SUBSYSTEM SHUTDOWN PROCESSING ***
FPMIMSAA   SHUTINIT,
             CMD=(,,'MSG ALL FPMIMSAA IS COMING DOWN')
*
FPMIMSAA   SHUTNORM,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,NORM')
*
FPMIMSAA   SHUTIMMED,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,IMMED')
*
FPMIMSAA   SHUTFORCE,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,FORCE')
*
************************************************************************
* BMP REGION 1 STATEMENTS
************************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  BMPIMSAA,JOB=BMPIMSAA,DESC='BATCH MESSAGE REGION - 1',
             PARENT=IMS10AA,SHUTDLY=00:01,STARTCMD=YES
*** SUBSYSTEM THRESHOLDS ***
THRESHOLDS BMPIMSAA,CRIT=(03,00:10),FREQ=(02,00:10),INFR=(01,00:10)
*** SUBSYSTEM AUTOMATION FLAGS ***
AUTOMATION BMPIMSAA,AUTO=Y
*** SUBSYSTEM FLAGS ***
BMPIMSAA   ENVIRON,
             SYNCH=Y,
             PRTNRSUB=BMPIMSBA,
             SUBTYPE=BMP
*** SUBSYSTEM START UP PROCESSING ***
BMPIMSAA   STARTUP,CMD=(,,'MVS S IMSRDR,MBR=BMPIMSAA')
*
*** SUBSYSTEM SHUTDOWN PROCESSING ***
BMPIMSAA   SHUTINIT,
             CMD=(,,'MSG ALL BMPIMSAA IS COMING DOWN')
*
BMPIMSAA   SHUTNORM,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,NORM')
*
BMPIMSAA   SHUTIMMED,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,IMMED')
*
BMPIMSAA   SHUTFORCE,
```

```
            CMD=(PASS1,,'EVIET00J &SUBSAPPL,FORCE')
*
***********************************************************************
* OTHER SUBSYSTEM DEFINITION FOR CHILD OF IMS CONTROL REGION          *
* THIS SUBSYSTEM BELONGS TO IMS CONTROL REGION BUT IS NOT AN          *
* IMS ELEMENT.                                                        *
***********************************************************************
SUBSYSTEM  XYZAA,JOB=XYZAA,DESC='XYZAA SUBSYSTEM CHILD OF IMS',
            PARENT=IMS10AA,STARTCMD=YES,PARMS='USER-PARMS',
            SHUTDLY=00:00:30
*
AUTOMATION XYZAA,AUTO=Y
*
THRESHOLDS  XYZAA,CRIT=(03,00:10),FREQ=(02,00:20),INFR=(01,00:30)
*
XYZAA      ENVIRON,
            SUBTYPE=OTHER
XYZAA      STARTUP,CMD=(,,'STARTUP CMD FOR SUBSYS XYZAA')
XYZAA      SHUTNORM,CMD=(PASS1,,'SHUTDOWN CMD FOR SUBSYS XYZAA')
*
*
***********************************************************************
*                            IMS10AB       B E G I N                  *
***********************************************************************
***********************************************************************
* IMS10AB STATEMENTS
***********************************************************************
SUBSYSTEM  IMS10AB,JOB=IMS10AB,DESC='TEST IMS SUBSYSTEM',
*           ARMNAME=IMS10,                                       @L1A
            PARENT=VTAM,SHUTDLY=00:20,STARTCMD=YES
*                                                                @L1A
*ASSOCIATION IMS10AB,                                            @L1A
*           SECONDARY=(SYS2,...,SYSn)                            @L1A
*
IMS10AB    STARTUP,CMD=(,,EVIEI00I)
*
*   IMS V2 AND V3 USE THE FOLLOWING START CMD FORMAT
*
*IMS10AB    START1,
*          CMD=(,,"MVS S IMS10AB,AUTO=&EHKVAR1,HSBID=&EHKVAR2")
*IMS10AB    START2,
*          CMD=(,,"MVS S IMS10AB,AUTO=&EHKVAR1,HSBID=&EHKVAR2")
*
*   IMS V4 USES THE FOLLOWING START CMD FORMAT
*
*IMS10AB    START1,
*     CMD=(,,"MVS S IMS10AB,PARM1='AUTO=&EHKVAR1,HSBID=&EHKVAR2'")
*IMS10AB    START2,
*     CMD=(,,"MVS S IMS10AB,PARM1='AUTO=&EHKVAR1,HSBID=&EHKVAR2'")
*
IMS10AB    ACORESTART,CMD=(,,'EVIEE00A IMS10AB')
IMSCNTL    IMS10AB,
            APPLID=(IMSXRF,IMS10AB),
            AUTOOPS=IMSOP03,
            APPLID1=IMS10AB,APPLID2=IMS10AB,
            MAJNODE1=,MAJNODE2=,
            OTHMAJ1=,OTHMAJ2=,
            PARMS1=,PARMS2=,
```

```
             DEFLTHSB=2
*
THRESHOLDS IMS10AB,CRIT=(2,08:00),FREQ=(1,04:00),INFR=(01,04:00)
*
AUTOMATION IMS10AB,AUTO=Y
INITSTART  IMS10AB,AUTO=E,EXITS=(EVIEIEXT)
START      IMS10AB,AUTO=Y
RECOVERY   IMS10AB,AUTO=Y
TERMINATE  IMS10AB,AUTO=Y
RESTART    IMS10AB,AUTO=E,EXITS=(EVIEIEXT)
IMS10AB    ENVIRON,
             SHUTGO=YES,
             ALTSTART=NO,
             SYNCH=Y,
             TKO=YES,
             DEFSTART=AUTO,
             DEFXRFST=XRF,
             ALTRST=NO,
             APPLVER=NO,
             PRTNRDOM=AC010,
             PRTNRID=IMSA,
             PRTNRSUB=IMS10AA,
             STRTTIMER=3,
             SUBID=IMSB,
             SHUTACT=NO,
             XRF=YES,
             SUBTYPE=CTL,
             WDELAY=20,
             SDELAY=50,
             BTIMER=3,
             ITIMER=9,
             RTIMER=3,
             TTIMER=3,
             WTCHNOTF=120,
             WTCHALERT=240
*** SUBSYSTEM MESSAGE AUTOMATION ***
IMS10AB    UNLKAVM,
             CMD=(,,'MVS R &EHKVAR1,UNLOCK')
IMS10AB    UNLOCK,
             REPLY=(,REPLY5,'/UNLOCK SYSTEM')
IMS10AB    RESTARTABORT,
                REPLY=(ERE,RETRY9,'/ERE.')                       /* @02C*/
*V2             REPLY=(OVERRIDE,RETRY9,'/ERE OVERRIDE.')
*V3             REPLY=(OVERRIDE,RETRY9,'/ERE COLDSYS OVERRIDE.')
IMS10AB    DFS3869A,
             REPLY=(,RETRY9,'/SWITCH SYSTEM FORCE')
IMS10AB    DFS3869I,
             REPLY=(,RETRY9,'/SWITCH SYSTEM FORCE')
IMS10AB    DFS810A,
             REPLY=(BACKUP,RETRY9,'/ERE BACKUP.'),
             REPLY=(BUILDQ,RETRY9,'/NRE FORMAT RS BUILDQ'),
             REPLY=(COLD,RETRY9,'/ERE BACKUP FORMAT SM QC LM SP.'),
             REPLY=(MANUAL,RETRY9,'&EHKVAR1')
IMS10AB    DFS994I,
             REPLY=(BUILDQ,RETRY9,'/STA DC'),
             REPLY=(BUILDQ,RETRY9,'/START PROGRAM EVISPPII'),
             REPLY=(BUILDQ,RETRY9,'/START TRAN EVITPPI1'),
             REPLY=(BUILDQ,RETRY9,'/CHE'),
```

```
             REPLY=(COLD,RETRY9,'/STA DC'),
             REPLY=(COLD,RETRY9,'/START PROGRAM EVISPPII'),
             REPLY=(COLD,RETRY9,'/START TRAN EVITPPI1'),
             REPLY=(COLD,RETRY9,'/CHE'),
             REPLY=(ERE,RETRY9,'/STA DC'),
             REPLY=(ERE,RETRY9,'/START PROGRAM EVISPPII'),
             REPLY=(ERE,RETRY9,'/START TRAN EVITPPI1'),
             REPLY=(ERE,RETRY9,'/ASSIGN LTERM APPLE TO NODE JONE'),
             REPLY=(ERE,RETRY9,'/CHE'),
             REPLY=(MANUAL,RETRY9,'/STA DC'),
             REPLY=(MANUAL,RETRY9,'/START PROGRAM EVISPPII'),
             REPLY=(MANUAL,RETRY9,'/START TRAN EVITPPI1'),
             REPLY=(MANUAL,RETRY9,'/CHE'),
             REPLY=(WARM,RETRY9,'/STA DC'),
             REPLY=(WARM,RETRY9,'/ASSIGN LTERM APPLE TO NODE JONE'),
             REPLY=(WARM,RETRY9,'/CHE'),
             REPLY=(WARM,RETRY9,'/START PROGRAM EVISPPII'),
             REPLY=(WARM,RETRY9,'/START TRAN EVITPPI1'),
             REPLY=(WARM,RETRY9,'/START TRAN ALL CLASS 6'),
             REPLY=(WARM,RETRY9,'/START DATABASE DTBSE1'),
             REPLY=(WARM,RETRY9,'/START LINE 4 PTERM 1, 2'),
             REPLY=(WARM,RETRY9,'/START LTERM APPLE, ORANG')
*
IMS10AB    PRECHKP,
             REPLY=(,RETRY9,'/PSTOP LINK ALL')
IMS10AB    POSTCHKP,
             REPLY=(,RETRY3,'/IDLE LINK ALL')
IMS10AB    STADC,
             REPLY=(,RETRY9,'/STA DC')
IMS10AB    HOLDQ,
             CMD=(,,'MVS $HQ,C=123')
IMS10AB    RELEASEQ,
             CMD=(,,'MVS $AQ,C=123')
IMS10AB    CHE,
             REPLY=(,RETRY9,'/CHE')
IMS10AB    SNAPQ,
             REPLY=(,RETRY9,'/CHE SNAPQ')
* NEW ENTRIES                                                       @01A*
IMS10AB    STOPREGION,
          REPLY=(NORMAL,RETRY9,'/STOP REG &EHKVAR1'),
          REPLY=(ABEND,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
          REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
          REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 CANCEL')
IMS10AB    STOPFPREGION,
          REPLY=(NORMAL,RETRY9,'/STOP REG &EHKVAR1'),
          REPLY=(ABEND,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
          REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
          REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 CANCEL')
IMS10AB    STOPBMPREGION,
          REPLY=(NORMAL,RETRY9,'/STOP REG &EHKVAR1'),
          REPLY=(ABEND,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
          REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
          REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 CANCEL')
* NEW ENTRIES                                                       @01A*
IMS10AB    VTAMTERMS,
             CMD=(SHUTDOWN,,'V NET,TERM,LU1=&EHKVAR1,TYPE=UNCOND'),
             CMD=(SHUTDOWN,,'V NET,ID=MTOCONSOLE,INACT'),
             CMD=(TAKEOVER,,'V NET,TERM,LU1=&EHKVAR1,TYPE=FORCE'),
```

```
            CMD=(TAKEOVER,,'V NET,INACT,ID=MTOCONSOLE,FORCE')
IMS10AB    FORCE,
            CMD=(,,'MVS F &EHKVAR1,STOP')
*
*   IMS SHUTDOWN COMMANDS
*
IMS10AB    BRO,
            REPLY=(,,'/BRO ACTIVE'),
            REPLY=(,RETRY9,'IMS COMING DOWN IN &EHKVAR1 MINUTES.')
IMS10AB    SHUTNORM,
            CMD=(PASS1,,'EVIET001 &SUBSAPPL,NORM,DUMPQ')
IMS10AB    SHUTIMMED,
            CMD=(PASS1,,'EVIET001 &SUBSAPPL,IMMED,FREEZE')
IMS10AB    SHUTFORCE,
            CMD=(PASS1,,'EVIET001 &SUBSAPPL,FORCE,NODUMP')
IMS10AB    SHUTTYPES,
            REPLY=(BACKUP,RETRY9,'/STOP BACKUP'),
            REPLY=(DUMPQ,RETRY9,'/CHE DUMPQ'),
            REPLY=(FREEZE,RETRY9,'/CHE FREEZE'),
            REPLY=(PURGE,RETRY9,'/CHE PURGE'),
            CMD=(NODUMP,,'MVS F &EHKVAR1,STOP'),
            CMD=(DUMP,,'MVS F &EHKVAR1,DUMP')
*** SUBSYSTEM ABEND CODES REQUIRED RESTART (NON-XRF OR NO-XRF) ***
IMS10AB    ABCODES,
            CODE=(SYS0C4,,,ABENDING),
            CODE=(IMS0113,,,ABENDING),
            CODE=(IMS0020,,,ABENDING),
            CODE=(IMS0707,,,ABENDING),
            CODE=(IMS0075,,,STOPPING)
IMS10AB    ACTCODES,
            CODE=(SYS0C4,,,ABENDING),
            CODE=(IMS0113,,,ABENDING),
            CODE=(IMS0020,,,ABENDING),
            CODE=(IMS0707,,,ABENDING),
            CODE=(IMS0604,,,ABENDING),
            CODE=(IMS0075,,,STOPPING)
IMS10AB    ALTCODES,
            CODE=(SYS0C4,,,ABENDING),
            CODE=(IMS0113,,,ABENDING),
            CODE=(IMS0020,,,ABENDING),
            CODE=(IMS0707,,,ABENDING),
            CODE=(IMS0075,,,STOPPING)
IMS10AB    TPABEND,
            CODE=(U0002,,,ABENDING)
*
* ENTRIES FOR RECOVERY OF IMS MSC LINKS
*
IMS10AB    DFS2161I,
            REPLY=(RESTART,,'/RSTART LINK &EHKVAR1')
IMS10AB    DFS2169I,
            REPLY=(RESTART,,'/RSTART LINK &EHKVAR1')
IMS10AB    DFS2142,
            REPLY=(START,,'/START MSNAME &EHKVAR2'),
            REPLY=(RESTART,,'/PSTOP LINK   &EHKVAR1'),
            REPLY=(RESTART,,'/START MSNAME &EHKVAR2'),
            REPLY=(RESTART,,'/RSTART LINK  &EHKVAR1')
*
RECOVERY   IMS10AB.MSC.0001,AUTO=N
```

```
*
* ENTRIES FOR RECOVERY AND ARCHIVE MONITORING OF OLDS
*
IMS10AB    OLDS,
             MINIMUM=03,
             SPARES=(05,99),
             ARCHIVETIME=00:20:00,
             RETRYCNT=5
IMS10AB    DFS3258A,
             CMD=(SYSTEM,,'IMSCMD &SUBSAPPL /LOG WAITING.'),
             CMD=(LAST,,'IMSCMD &SUBSAPPL /STA OLDS 99.')
*
THRESHOLDS IMS10AB.OLDS,
             CRIT=(05,00:30),FREQ=(03,00:30),INFR=(02,00:30)
RECOVERY   IMS10AB.OLDS,AUTO=Y
*
* ENTRIES FOR RECON RECOVERY
*
IMS10AA    RECONS,
             MONITOR=00:15:00
*
* ENTRIES FOR TRANSACTION/PROGRAM RECOVERY
*
IMS10AB    DFS554A,
             CMD=(TRAN,,'IMSCMD &SUBSAPPL /STA TRAN &EHKVAR1'),
             CMD=(PROG,,'IMSCMD &SUBSAPPL /STA PGM &EHKVAR2')
RECOVERY   IMS10AB.DFS554A,AUTO=Y
*     CONTROLLING TRANACTION RECOVERY FOR:
*       1) MESSAGE PROCESSING REGIONS
*       2) FASTPATH REGIONS
*       3) TRANSACTION DRIVEN BMP REGIONS
ABCODETRAN IMS10AB.TRAN,
             CODE=(SAMPLE1,U0778,*,EXCLUDE),
             CODE=(*,*,*,INCLUDE)
THRESHOLDS IMS10AB.TRAN,
             CRIT=(09,00:15),FREQ=(08,00:25),INFR=(04,00:35)
THRESHOLDS IMS10AB.TRAN.SAMPLE2,
             CRIT=(05,00:05),FREQ=(04,00:05),INFR=(03,00:05)
RECOVERY   IMS10AB.TRAN,AUTO=Y
RECOVERY   IMS10AB.TRAN.SAMPLE3,AUTO=Y
*     CONTROLLING PROGRAM RECOVERY
ABCODEPROG IMS10AB.PROG,
             CODE=(*,U0778,SAMPLE4,EXCLUDE),
             CODE=(*,*,*,INCLUDE)
THRESHOLDS IMS10AB.PROG,
             CRIT=(09,00:15),FREQ=(08,00:25),INFR=(04,00:35)
THRESHOLDS IMS10AB.PROG.SAMPLE5,
             CRIT=(05,00:05),FREQ=(04,00:05),INFR=(03,00:05)
RECOVERY   IMS10AB.PROG,AUTO=Y
RECOVERY   IMS10AB.PROG.SAMPLE6,AUTO=Y
RECOVERY   IMS10AB.PROG.SAMPLE7,AUTO=N
*
* ENTRIES FOR USER DEFINED RESTART COMMANDS
*
IMS10AB    USERSTART,
             ERE1='ERE COLDCOMM',
             NRE1='NRE CHKPT 0'
*
```

```
* ENTRIES FOR SERVICE PERIODS
*
SERVICE    IMS10AB,
             DAY=(DAILY,0745-1715),
             DAY=(WEEKEND,DOWN-DOWN)
*
* ENTRIES FOR TRIGGERS
*
TRIGGER    IMS10AB,
             STARTUP=(SERVICE,JOB1DONE),
             SHUTDOWN=DOWN1
*
*
* ENTRIES FOR EXTCOND
*
EXTCOND    JOB1DONE,
             UNSET=START,
             DESC='IMS JOB 1 MUST BE COMPLETE'
*
* SAMPLE ENTRIES FOR THE TCO OPERATOR INTERFACE
*    - TCO IS ONLY SUPPORTED ON IMS V3 OR HIGHER                 @03C*
*
*V3 IMS10AB   TCO,
*V3             REPLY=(INIT,,'DFSTCF LOAD DFSTCF  .'),
*V3             REPLY=(SPEC,,'DFSTCF LOAD &EHKVAR1 .'),          @03C*
*V3             REPLY=(START,,'/START LTERM DFSTCFI .'),
*V3             REPLY=(STOP,,'/PSTOP LTERM DFSTCFI .')
************************************************************************
* DBRC STATEMENTS
************************************************************************
SUBSYSTEM  DBRIMSAB,JOB=DBRIMSAB,DESC='DBRC SUBSYSTEM OF IMS10AB',
             PARENT=IMS10AB,SHUTDLY=00:01,SHUTOPTIONS=PARENT,
             STARTOPTIONS=PARENT
*
THRESHOLDS DBRIMSAB,CRIT=(03,00:10),FREQ=(02,00:10),INFR=(01,00:10)
*
AUTOMATION DBRIMSAB,AUTO=Y
DBRIMSAB   ENVIRON,
             SUBTYPE=DBRC
************************************************************************
* DLISAS STATEMENTS
************************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  DLIIMSAB,JOB=DLIIMSAB,DESC='DLISAS SUBSYSTEM OF IMS10AB',
             PARENT=IMS10AB,SHUTDLY=00:01,SHUTOPTIONS=PARENT,
             STARTOPTIONS=PARENT
*
THRESHOLDS DLIIMSAB,CRIT=(03,00:10),FREQ=(02,00:10),INFR=(01,00:10)
*
AUTOMATION DLIIMSAB,AUTO=Y
*** SUBSYSTEM FLAGS ***
DLIIMSAB   ENVIRON,
             SUBTYPE=DLS
************************************************************************
* MSG REGION 1 STATEMENTS
************************************************************************
SUBSYSTEM  MSGIMSBA,JOB=MSGIMSBA,DESC='ONLINE MSG REGION 1',
             PARENT=IMS10AB,SHUTDLY=00:01,
```

```
            SHUTOPTIONS=PARENT,
            STARTOPTIONS=PARENT
THRESHOLDS MSGIMSBA,CRIT=(03,00:10),FREQ=(02,00:10),INFR=(01,00:10)
AUTOMATION MSGIMSBA,AUTO=Y
MSGIMSBA   ENVIRON,
            SYNCH=Y,
            PRTNRSUB=MSGIMSAA,
            SUBTYPE=TP
MSGIMSBA   START1,
            CMD=(,,'MVS S IMSRDR,MBR=MSGIMSBA')
MSGIMSBA   START2,
            CMD=(,,'MVS S IMSRDR,MBR=MSGIMSBA')
*********************************************************************
SUBSYSTEM  MSGIMSBB,JOB=MSGIMSBB,DESC='ONLINE MSG REGION 2',
            PARENT=IMS10AB,SHUTDLY=00:01,STARTCMD=YES
THRESHOLDS MSGIMSBB,CRIT=(03,00:10),FREQ=(02,00:10),INFR=(01,00:10)
AUTOMATION MSGIMSBB,AUTO=Y
MSGIMSBB   ENVIRON,
            SYNCH=Y,
            SUBTYPE=TP,
            PRTNRSUB=MSGIMSAB
MSGIMSBB   STARTUP,
            CMD=(,,'MVS S IMSRDR,MBR=MSGIMSBB')
MSGIMSBB   START1,
            CMD=(,,'MVS S IMSRDR,MBR=MSGIMSBB')
MSGIMSBB   START2,
            CMD=(,,'MVS S IMSRDR,MBR=MSGIMSBB')
***  SUBSYSTEM SHUTDOWN PROCESSING                                 ***
MSGIMSBB   SHUTINIT,
            CMD=(,,'MSG ALL MSGIMSBB IS COMING DOWN')
MSGIMSBB   SHUTNORM,
            CMD=(PASS1,,'EVIET00J &SUBSAPPL,NORM')
MSGIMSBB   SHUTIMMED,
            CMD=(PASS1,,'EVIET00J &SUBSAPPL,IMMED')
MSGIMSBB   SHUTFORCE,
            CMD=(PASS1,,'EVIET00J &SUBSAPPL,FORCE')
*********************************************************************
* FPM REGION 1 STATEMENTS
*********************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  FPMIMSBA,JOB=FPMIMSBA,DESC='FAST PATH REGION 1',
            PARENT=IMS10AB,SHUTDLY=00:01,STARTCMD=YES
*** SUBSYSTEM THRESHOLDS ***
THRESHOLDS FPMIMSBA,CRIT=(03,00:10),FREQ=(02,00:10),INFR=(01,00:10)
*** SUBSYSTEM AUTOMATION FLAGS ***
AUTOMATION FPMIMSBA,AUTO=Y
*** SUBSYSTEM FLAGS ***
FPMIMSBA   ENVIRON,
            SYNCH=Y,
            PRTNRSUB=FPMIMSAA,
            SUBTYPE=FP
*** SUBSYSTEM START UP PROCESSING ***
FPMIMSBA   STARTUP,CMD=(,,'MVS S IMSRDR,MBR=FPMIMSBA')
*
*** SUBSYSTEM SHUTDOWN PROCESSING ***
FPMIMSBA   SHUTINIT,
```

```
             CMD=(,,'MSG ALL FPMIMSAA IS COMING DOWN')
*
FPMIMSBA   SHUTNORM,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,NORM')
*
FPMIMSBA   SHUTIMMED,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,IMMED')
*
FPMIMSBA   SHUTFORCE,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,FORCE')
*
***********************************************************************
* BMP REGION 1 STATEMENTS
***********************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  BMPIMSBA,JOB=BMPIMSBA,DESC='BATCH MESSAGE REGION - 1',
             PARENT=IMS10AB,SHUTDLY=00:01,STARTCMD=YES
*** SUBSYSTEM THRESHOLDS ***
THRESHOLDS BMPIMSBA,CRIT=(03,00:10),FREQ=(02,00:10),INFR=(01,00:10)
*** SUBSYSTEM AUTOMATION FLAGS ***
AUTOMATION BMPIMSBA,AUTO=Y
*** SUBSYSTEM FLAGS ***
BMPIMSBA   ENVIRON,
             SYNCH=Y,
             PRTNRSUB=BMPIMSAA,
             SUBTYPE=BMP
*** SUBSYSTEM START UP PROCESSING ***
BMPIMSBA   STARTUP,CMD=(,,'MVS S IMSRDR,MBR=BMPIMSBA')
*
*** SUBSYSTEM SHUTDOWN PROCESSING ***
BMPIMSBA   SHUTINIT,
             CMD=(,,'MSG ALL BMPIMSAA IS COMING DOWN')
*
BMPIMSBA   SHUTNORM,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,NORM')
*
BMPIMSBA   SHUTIMMED,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,IMMED')
*
BMPIMSBA   SHUTFORCE,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,FORCE')
*
***********************************************************************
* OTHER SUBSYSTEM DEFINITION FOR CHILD OF IMS CONTROL REGION          *
* THIS SUBSYSTEM BELONGS TO IMS CONTROL REGION BUT IS NOT AN          *
* IMS ELEMENT.                                                        *
***********************************************************************
SUBSYSTEM  XYZAB,JOB=XYZAB,DESC='XYZAA SUBSYSTEM CHILD OF IMS',
             PARENT=IMS10AB,STARTCMD=YES,PARMS='USER-PARMS',
             SHUTDLY=00:00:30
*
AUTOMATION XYZAB,AUTO=Y
*
THRESHOLDS  XYZAB,CRIT=(03,00:10),FREQ=(02,00:20),INFR=(01,00:30)
*
XYZAB      ENVIRON,
             SUBTYPE=OTHER
XYZAB      STARTUP,CMD=(,,'STARTUP CMD FOR SUBSYS XYZAB')
```

```
XYZAB      SHUTNORM,CMD=(PASS1,,'SHUTDOWN CMD FOR SUBSYS XYZAB')
*
************************************************************************
*              XRF IMS ENTRIES (IMS10AA - IMS10BB)  E N D              *
************************************************************************
```

EVIS2SPX

```
*********************************************************************
*                                                                   *
*       Licensed Materials - Property of IBM                        *
*       5685-151                                                    *
*       (C) Copyright IBM CORP. 1990, 1999                          *
*                                                                   *
*********************************************************************
*     APAR#      Date   By Description                              *
*  ----------- -------- -- ---------------------------------------- *
*  $09=OW35606,V1R4,11Jan99,APC(NS) : Cater for SAfOS390 and IMS V6   *
*                                  utilizing CQS & FDR              *
*                                                                   *
*********************************************************************
* STATUS FILE ENTRIES : DURING INITIALIZATION, IMS EXIT WILL READ
*                       AND ALLOCATE, AND START THE DST TASKS
*
*
*********************************************************************
*          IMS ENTRIES (IMS10S) Utilizing CQS & FDR         B E G I N
*********************************************************************
* NOTE: The following entries flagged with @FDR are the required entries
*       to merge with existing ACFs to utilize  CQS & FDR.
*       BUILDQ replies are no longer required and have been removed .
*********************************************************************
* IMS10S STATEMENTS
*********************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  IMS10S,
             JOB=IMS10SJ,
             DESC='TEST IMS SUBSYSTEM',
*            ARMNAME=IMS10,         ARM ELEMENT NAME IF APPLICABLE
             PARENT=VTAM,
             SHUTDLY=00:20,
             STARTCMD=YES
*
*ASSOCIATION IMS10S,
*              SECONDARY=(SYS2,...,SYSn)
*
IMS10S     STARTUP,
             CMD=(,,EVIEI00I)
*                               Merge the following 2 lines ------->@FDR
IMS10S     UP,
             CMD=(,,EVISTFDR)
*
*   IMS V2 AND V3 USE THE FOLLOWING START CMD FORMAT
*
*IMS10S     START1,
*             CMD=(,,"MVS S IMS10SJ,AUTO=&EHKVAR1")
*
*   IMS V4 USES THE FOLLOWING START CMD FORMAT
*
*IMS10S     START1,
*             CMD=(,,"MVS S IMS10SJ,PARM1='AUTO=&EHKVAR1'")
*
IMSCNTL    IMS10S,
             APPLID=IMS10S,
```

```
              AUTOOPS=IMSOP01,
              APPLID1=IMS10S
*
THRESHOLDS IMS10S,
              CRIT=(2,08:00),
              FREQ=(1,04:00),
              INFR=(01,4:00)
*
IMS10S     ACORESTART,CMD=(,,'EVIEE00A IMS10S')
*
AUTOMATION IMS10S,
              AUTO=Y
INITSTART  IMS10S,
              AUTO=E,
              EXITS=(EVIEIEXT)
START      IMS10S,
              AUTO=Y
RECOVERY   IMS10S,
              AUTO=Y
TERMINATE  IMS10S,
              AUTO=Y
RESTART    IMS10S,
              AUTO=E,
              EXITS=(EVIEIEXT)
*** SUBSYSTEM FLAGS ***
IMS10S     ENVIRON,
              SUBID=IMSZ,
              XRF=NO,
              DEFSTART=AUTO,
              SUBTYPE=CTL,
              BTIMER=3,
              ITIMER=6,
              RTIMER=3,
              STRTTIMER=3,
*                                Merge the following 2 lines ------->@FDR
              PRTNRDOM=AOC10,
              PRTNRSUB=IMS10F,
              TTIMER=3
*** SUBSYSTEM MESSAGE AUTOMATION ***
IMS10S     RESTARTABORT,
              REPLY=(ERE,RETRY9,'/ERE.')
*V2           REPLY=(OVERRIDE,RETRY9,'/ERE OVERRIDE.')
*V3           REPLY=(OVERRIDE,RETRY9,'/ERE COLDSYS OVERRIDE.')
IMS10S     DFS810A,
              REPLY=(AUTO,RETRY9,'/NRE.'),
*V2           REPLY=(COLD,RETRY9,'/NRE CHKPT 0 FORMAT ALL DETACH'),
*V3           REPLY=(COLD,RETRY9,'/NRE CHKPT 0 FORMAT ALL'),
              REPLY=(MANUAL,RETRY9,'&EHKVAR1')
IMS10S     DFS994I,
              REPLY=(COLD,RETRY9,'/STA DC'),
              REPLY=(COLD,RETRY9,'/START PROGRAM EVISPPII'),
              REPLY=(COLD,RETRY9,'/START TRAN EVITPPI1'),
              REPLY=(COLD,RETRY9,'/CHE'),
              REPLY=(ERE,RETRY9,'/STA DC'),
              REPLY=(ERE,RETRY9,'/START PROGRAM EVISPPII'),
              REPLY=(ERE,RETRY9,'/START TRAN EVITPPI1'),
              REPLY=(ERE,RETRY9,'/ASSIGN LTERM APPLE TO NODE JONE'),
              REPLY=(ERE,RETRY9,'/CHE'),
```

```
              REPLY=(MANUAL,RETRY9,'/STA DC'),
              REPLY=(MANUAL,RETRY9,'/START PROGRAM EVISPPII'),
              REPLY=(MANUAL,RETRY9,'/START TRAN EVITPPI1'),
              REPLY=(MANUAL,RETRY9,'/CHE'),
              REPLY=(WARM,RETRY9,'/STA DC'),
              REPLY=(WARM,RETRY9,'/START PROGRAM EVISPPII'),
              REPLY=(WARM,RETRY9,'/START TRAN EVITPPI1'),
              REPLY=(WARM,RETRY9,'/ASSIGN CPRI 8 TO TRAN PIT, SEED'),
              REPLY=(WARM,RETRY9,'/CHE'),
              REPLY=(WARM,RETRY9,'/START TRAN ALL CLASS 6'),
              REPLY=(WARM,RETRY9,'/START DATABASE DTBSE1'),
              REPLY=(WARM,RETRY9,'/START LINE 4 PTERM 1, 2'),
              REPLY=(WARM,RETRY9,'/START LTERM APPLE, ORANG')
*
IMS10S     PRECHKP,
              REPLY=(,RETRY9,'/PSTOP LTERM APPLE, ORANG'),
              REPLY=(,RETRY9,'/PSTOP LINK ALL')
IMS10S     POSTCHKP,
              REPLY=(,RETRY9,'/IDLE NODE ABC'),
              REPLY=(,RETRY9,'/IDLE LINK 2')
IMS10S     STADC,
              REPLY=(,RETRY9,'/STA DC')
IMS10S     HOLDQ,
              CMD=(,,'MVS $HQ,C=123')
IMS10S     RELEASEQ,
              CMD=(,,'MVS $AQ,C=123')
IMS10S     CHE,
              REPLY=(,RETRY9,'/CHE')
IMS10S     SNAPQ,
              REPLY=(,RETRY9,'/CHE SNAPQ')
*
IMS10S     STOPREGION,
              REPLY=(NORMAL,RETRY9,'/STOP REG &EHKVAR1'),
              REPLY=(ABEND,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
              REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
              REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 CANCEL')
IMS10S     STOPFPREGION,
              REPLY=(NORMAL,RETRY9,'/STOP REG &EHKVAR1'),
              REPLY=(ABEND,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
              REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
              REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 CANCEL')
IMS10S     STOPBMPREGION,
              REPLY=(NORMAL,RETRY9,'/STOP REG &EHKVAR1'),
              REPLY=(ABEND,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
              REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 ABDUMP &EHKVAR2'),
              REPLY=(CANCEL,RETRY9,'/STOP REG &EHKVAR1 CANCEL')
*
IMS10S     VTAMTERMS,
              CMD=(SHUTDOWN,,'V NET,TERM,LU1=&EHKVAR1,TYPE=UNCOND'),
              CMD=(SHUTDOWN,,'V NET,INACT,ID=MTOCONS,FORCE')
*
*   IMS SHUTDOWN COMMANDS
*
IMS10S     BRO,
              REPLY=(,,'/BRO ACTIVE'),
              REPLY=(,RETRY9,'IMS COMING DOWN IN &EHKVAR1 MINUTES.')
IMS10S     SHUTNORM,
```

```
             CMD=(PASS1,,'EVIET001 &SUBSAPPL,NORM,DUMPQ')
IMS10S     SHUTIMMED,
             CMD=(PASS1,,'EVIET001 &SUBSAPPL,IMMED,FREEZE')
IMS10S     SHUTFORCE,
             CMD=(PASS1,,'EVIET001 &SUBSAPPL,FORCE,NODUMP')
IMS10S     SHUTTYPES,
             REPLY=(DUMPQ,RETRY9,'/CHE DUMPQ NOCQSSHUT'),
             REPLY=(FREEZE,RETRY9,'/CHE FREEZE'),
             REPLY=(PURGE,RETRY9,'/CHE PURGE'),
             CMD=(NODUMP,,'MVS F &EHKVAR1,STOP'),
             CMD=(DUMP,,'MVS F &EHKVAR1,DUMP')
*** SUBSYSTEM ABEND CODES REQUIRED RESTART (NON-XRF OR NO-XRF) ***
IMS10S     ABCODES,
             CODE=(SYS0C4,,,ABENDING),
             CODE=(IMS0113,,,ABENDING),
             CODE=(IMS0020,,,ABENDING),
             CODE=(IMS0707,,,ABENDING),
             CODE=(IMS0075,,,STOPPING),
             CODE=(*,,,STOPPING)
IMS10S     TPABEND,
             CODE=(U0002,,,ABENDING)
*
* ENTRIES FOR RECOVERY OF IMS MSC LINKS
*
IMS10S     DFS2161I,
             REPLY=(RESTART,,'/RSTART LINK &EHKVAR1')
IMS10S     DFS2169I,
             REPLY=(RESTART,,'/RSTART LINK &EHKVAR1')
IMS10S     DFS2142,
             REPLY=(START,,'/START MSNAME &EHKVAR2'),
             REPLY=(RESTART,,'/PSTOP LINK   &EHKVAR1'),
             REPLY=(RESTART,,'/START MSNAME &EHKVAR2'),
             REPLY=(RESTART,,'/RSTART LINK  &EHKVAR1')
*
RECOVERY   IMS10S.MSC.0001,
             AUTO=Y
*
* ENTRIES FOR RECOVERY AND ARCHIVE MONITORING OF IMS OLDS
*
IMS10S     OLDS,
             MINIMUM=03,
             SPARES=(05,99),
             ARCHIVETIME=00:20:00,
             RETRYCNT=5
*
* ENTRIES FOR CQSET Command for shared queue           -------> @FDR
*
*
IMS10S     DFS3258A,
             CMD=(SYSTEM,,'IMSCMD &SUBSAPPL /LOG WAITING.'),
             CMD=(LAST,,'IMSCMD &SUBSAPPL /STA OLDS 99.')
*
THRESHOLDS IMS10S.OLDS,
             CRIT=(05,00:30),
             FREQ=(03,00:30),
             INFR=(02,00:30)
RECOVERY   IMS10S.OLDS,
```

```
              AUTO=Y
*
* ENTRIES FOR RECON RECOVERY
*
IMS10S     RECONS,
              MONITOR=00:15:00
*
* ENTRIES FOR TRANSACTION/PROGRAM RECOVERY
*
IMS10S     DFS554A,
              CMD=(TRAN,,'IMSCMD &SUBSAPPL /STA TRAN &EHKVAR1'),
              CMD=(PROG,,'IMSCMD &SUBSAPPL /STA PGM &EHKVAR2')
*
RECOVERY   IMS10S.DFS554A,AUTO=Y
*     CONTROLLING TRANACTION RECOVERY FOR:
*       1) MESSAGE PROCESSING REGIONS
*       2) FASTPATH REGIONS
*       3) TRANSACTION DRIVEN BMP REGIONS
ABCODETRAN IMS10S.TRAN,
              CODE=(SAMPLE1,U0778,*,EXCLUDE),
              CODE=(*,*,*,INCLUDE)
THRESHOLDS IMS10S.TRAN,
              CRIT=(09,00:15),
              FREQ=(06,00:20),
              INFR=(03,00:35)
THRESHOLDS IMS10S.TRAN.SAMPLE2,
              CRIT=(05,00:05),
              FREQ=(04,00:15),
              INFR=(03,00:25)
THRESHOLDS IMS10S.TRAN.SAMPLE3,
              CRIT=(05,00:05),
              FREQ=(04,00:15),
              INFR=(03,00:25)
RECOVERY   IMS10S.TRAN,
              AUTO=Y
RECOVERY   IMS10S.TRAN.SAMPLE2,
              AUTO=Y
RECOVERY   IMS10S.TRAN.SAMPLE3,
              AUTO=Y
RECOVERY   IMS10S.TRAN.SAMPLE4,
              AUTO=Y
*     CONTROLLING PROGRAM RECOVERY
ABCODEPROG IMS10S.PROG,
              CODE=(*,U0778,SAMPLE4,EXCLUDE),
              CODE=(*,*,*,INCLUDE)
THRESHOLDS IMS10S.PROG,
              CRIT=(09,00:15),
              FREQ=(07,00:25),
              INFR=(04,00:35)
THRESHOLDS IMS10S.PROG.SAMPLE5,
              CRIT=(05,00:05),
              FREQ=(04,00:05),
              INFR=(03,00:05)
RECOVERY   IMS10S.PROG,
              AUTO=Y
RECOVERY   IMS10S.PROG.SAMPLE6,
              AUTO=Y
*
```

```
* ENTRIES FOR USER DEFINED RESTART COMMANDS
*
IMS10S     USERSTART,
             ERE1='ERE COLDCOMM',
             NRE1='NRE CHKPT 0'
*
* ENTRIES FOR SERVICE PERIODS
*
SERVICE    IMS10S,
             DAY=(DAILY,0745-1715),
             DAY=(WEEKEND,DOWN-DOWN)
*
* ENTRIES FOR TRIGGERS
*
TRIGGER    IMS10S,
             STARTUP=(SERVICE,IMSJOB1),
             SHUTDOWN=IMSJOB2
*
* ENTRIES FOR EXTCOND
*
EXTCOND    IMSJOB1,
             UNSET=START,
             DESC='IMS JOB 1 MUST BE COMPLETE'
*
* SAMPLE ENTRIES FOR THE TCO OPERATOR INTERFACE
*    - TCO IS ONLY SUPPORTED ON IMS V3                                  *
*
*V3 IMS10S    TCO,
*V3             REPLY=(INIT,,'DFSTCF LOAD DFSTCF  .'),
*V3             REPLY=(SPEC,,'DFSTCF LOAD &EHKVAR1 .'),
*V3             REPLY=(START,,'/START LTERM DFSTCFI .'),
*V3             REPLY=(STOP,,'/PSTOP LTERM DFSTCFI .')
*************************************************************************
* DBRC STATEMENTS
*************************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  DBRIMSS,
             JOB=DBRIMSS,
             DESC='DBRC SUBSYSTEM OF IMS10S',
             PARENT=IMS10S,
             SHUTDLY=00:01,
             SHUTOPTIONS=PARENT,
             STARTOPTIONS=PARENT
*
THRESHOLDS DBRIMSS,
             CRIT=(03,00:10),
             FREQ=(02,00:10),
             INFR=(01,00:10)
*
AUTOMATION DBRIMSS,
             AUTO=Y
*** SUBSYSTEM FLAGS ***
DBRIMSS    ENVIRON,
             SUBTYPE=DBRC
*************************************************************************
* DLISAS STATEMENTS
*************************************************************************
```

```
*** SUBSYSTEM NAME ***
SUBSYSTEM  DLIIMSS,
             JOB=DLIIMSS,
             DESC='DLISAS SUBSYSTEM OF IMS10S'
             PARENT=IMS10S,
             SHUTDLY=00:01,
             SHUTOPTIONS=PARENT,
             STARTOPTIONS=PARENT
*
THRESHOLDS DLIIMSS,
             CRIT=(03,00:10),
             FREQ=(02,00:10),
             INFR=(01,00:10)
*
AUTOMATION DLIIMSS,
             AUTO=Y
*** SUBSYSTEM FLAGS ***
DLIIMSS    ENVIRON,
             SUBTYPE=DLS
************************************************************************
* MSG REGION 1 STATEMENTS
************************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  MSGIMSSA,
             JOB=MSGIMSSA,
             DESC='ONLINE MSG REGION 1',
             PARENT=IMS10S,
             SHUTDLY=00:01,
             SHUTOPTIONS=PARENT,
             STARTOPTIONS=PARENT
*** SUBSYSTEM THRESHOLDS ***
THRESHOLDS MSGIMSSA,
             CRIT=(03,00:10),
             FREQ=(02,00:10),
             INFR=(01,00:10)
*** SUBSYSTEM AUTOMATION FLAGS ***
AUTOMATION MSGIMSSA,
             AUTO=Y
*** SUBSYSTEM FLAGS ***
MSGIMSSA   ENVIRON,
             SUBTYPE=TP
*** SUBSYSTEM START UP PROCESSING ***
MSGIMSSA   START1,
             CMD=(,,'MVS S IMSRDR,MBR=MSGIMSSA')
*
************************************************************************
* MSG REGION 2 STATEMENTS
************************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  MSGIMSSB,
             JOB=MSGIMSSB,
             DESC='ONLINE MSG REGION 2',
             PARENT=IMS10S,
             SHUTDLY=00:01,
             STARTCMD=YES
*** SUBSYSTEM THRESHOLDS ***
THRESHOLDS MSGIMSSB,
             CRIT=(03,00:10),
```

```
              FREQ=(02,00:10),
              INFR=(01,00:10)
*** SUBSYSTEM AUTOMATION FLAGS ***
AUTOMATION MSGIMSSB,
              AUTO=Y
*** SUBSYSTEM FLAGS ***
MSGIMSSB   ENVIRON,
              SUBTYPE=TP
*** SUBSYSTEM START UP PROCESSING ***
MSGIMSSB   STARTUP,
              CMD=(,,'MVS S IMSRDR,MBR=MSGIMSSB')
*
****************  SUBSYSTEM START UP OPTION AVAILABLE FOR V6
*MSGIMSSB   STARTUP,
*             CMD=(,,'IMSCMD IMS10S /STA REG IMS01MPR JOBNAME MSGIMSSB')
*
*** SUBSYSTEM SHUTDOWN PROCESSING ***
MSGIMSSB   SHUTINIT,
              CMD=(,,'MSG ALL MSGIMSSB IS COMING DOWN')
*
MSGIMSSB   SHUTNORM,
              CMD=(PASS1,,'EVIET00J &SUBSAPPL,NORM')
*
MSGIMSSB   SHUTIMMED,
              CMD=(PASS1,,'EVIET00J &SUBSAPPL,IMMED')
*
MSGIMSSB   SHUTFORCE,
              CMD=(PASS1,,'EVIET00J &SUBSAPPL,FORCE')
*
***********************************************************************
* FPM REGION 1 STATEMENTS
***********************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  FPMIMSSA,
              JOB=FPMIMSSA,
              DESC='FAST PATH REGION 1',
              PARENT=IMS10S,
              SHUTDLY=00:01,
              STARTCMD=YES
*** SUBSYSTEM THRESHOLDS ***
THRESHOLDS FPMIMSSA,
              CRIT=(03,00:10),
              FREQ=(02,00:10),
              INFR=(01,00:10)
*** SUBSYSTEM AUTOMATION FLAGS ***
AUTOMATION FPMIMSSA,
              AUTO=Y
*** SUBSYSTEM FLAGS ***
FPMIMSSA   ENVIRON,
              SUBTYPE=FP
*** SUBSYSTEM START UP PROCESSING ***
FPMIMSSA   STARTUP,
              CMD=(,,'MVS S IMSRDR,MBR=FPMIMSSA')
*
*** SUBSYSTEM SHUTDOWN PROCESSING ***
FPMIMSSA   SHUTINIT,
```

```
             CMD=(,,'MSG ALL FPMIMSSA IS COMING DOWN')
*
FPMIMSSA   SHUTNORM,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,NORM')
*
FPMIMSSA   SHUTIMMED,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,IMMED')
*
FPMIMSSA   SHUTFORCE,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,FORCE')
*
************************************************************************
* BMP REGION 1 STATEMENTS
************************************************************************
*** SUBSYSTEM NAME ***
SUBSYSTEM  BMPIMSSA,
             JOB=BMPIMSSA,
             DESC='BATCH MESSAGE REGION - 1',
             PARENT=IMS10S,
             SHUTDLY=00:01,
             STARTCMD=YES
*** SUBSYSTEM THRESHOLDS ***
THRESHOLDS BMPIMSSA,
             CRIT=(03,00:10),
             FREQ=(02,00:10),
             INFR=(01,00:10)
*** SUBSYSTEM AUTOMATION FLAGS ***
AUTOMATION BMPIMSSA,AUTO=Y
*** SUBSYSTEM FLAGS ***
BMPIMSSA   ENVIRON,
             SUBTYPE=BMP
*** SUBSYSTEM START UP PROCESSING ***
BMPIMSSA   STARTUP,
             CMD=(,,'MVS S IMSRDR,MBR=BMPIMSSA')
*
*** SUBSYSTEM SHUTDOWN PROCESSING ***
BMPIMSSA   SHUTINIT,
             CMD=(,,'MSG ALL BMPIMSSA IS COMING DOWN')
*
BMPIMSSA   SHUTNORM,
             CMD=(PASS1,,'user - supplied shutdown command')
*
BMPIMSSA   SHUTIMMED,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,IMMED')
*
BMPIMSSA   SHUTFORCE,
             CMD=(PASS1,,'EVIET00J &SUBSAPPL,FORCE')
*
************************************************************************
* OTHER SUBSYSTEM DEFINITION FOR CHILD OF IMS CONTROL REGION           *
* THIS SUBSYSTEM BELONGS TO IMS CONTROL REGION BUT IS NOT AN           *
* IMS ELEMENT.                                                         *
************************************************************************
SUBSYSTEM  ABCDE,
             JOB=ABCDE,
             DESC='ABCDE SUBSYSTEM CHILD OF IMS',
             PARENT=IMS10S,
             STARTCMD=YES,
```

```
              PARMS='USER-PARMS',
              SHUTDLY=00:00:30
*
AUTOMATION ABCDE,
              AUTO=Y
*
THRESHOLDS  ABCDE,CRIT=(03,00:10),FREQ=(02,00:20),INFR=(01,00:30)
*
ABCDE       ENVIRON,
              SUBTYPE=OTHER
ABCDE       STARTUP,
              CMD=(,,'STARTUP CMD FOR SUBSYS ABCDE')
ABCDE       SHUTNORM,
              CMD=(PASS1,,'SHUTDOWN CMD FOR SUBSYS ABCDE')
*
*
************************************************************************
*              NON-XRF IMS ENTRIES (IMS10S)     E N D                  *
************************************************************************
```

EVIS2FDR

```
*********************************************************************
* (C) COPYRIGHT IBM CORP. 1990, 1999 - ALL RIGHTS RESERVED          *
*********************************************************************
*                                                                   *
*        LICENSED MATERIALS - PROPERTY OF IBM                       *
*        RESTRICTED MATERIALS OF IBM                                *
*        5685-151                                                   *
*        (C) COPYRIGHT IBM CORP. 1990, 1999                         *
*                                                                   *
*        US GOVERNMENT USERS RESTRICTED RIGHTS -                    *
*        USE, DUPLICATION OR DISCLOSURE RESTRICTED BY               *
*        GSA ADP SCHEDULE CONTRACT WITH IBM CORP.                   *
*                                                                   *
*********************************************************************
* DESCRIPTION: AOC/MVS CONFIGURATION CONTROL FILE FOR IMS PARTNER   *
*              FDR DEFINITIONS.                                     *
*********************************************************************
*   APAR#     DATE                                                  *
*  -------  --------                                                *
*  $L1=OW35606,V1R4,17Nov98,APC(NS):New sample for FDR processing.  *
*********************************************************************
*********************************************************************
*   ACF entries for FDR                                             *
*********************************************************************
*
SUBSYSTEM  IMS10F,
           DESC='FDR for IMS SUBSYSTEM',
           JOBTYPE=MVS,
           JOB=IMS10F,
           PARENT=(JES2),
           STRTDLY=00:05:00,
           RESTARTOPT=ABENDONLY
*
*
* SHUTNORM COMMAND
*
IMS10F SHUTNORM,
      CMD=(PASS1,,'MVS F &SUBSJOB,TERM')
*
*
* SHUTIMMED COMMAND
*
IMS10F SHUTIMMED,
      CMD=(PASS1,,'MVS F &SUBSJOB,STOP')
*
*
* DSF4167A message reply
*
IMS10F AVM006E,
      REPLY=(,RETRY05,'UNLOCK')
```

EVIS2CQS

```
**********************************************************************
* (C) COPYRIGHT IBM CORP. 1990, 1999 - ALL RIGHTS RESERVED           *
**********************************************************************
*                                                                    *
*        LICENSED MATERIALS - PROPERTY OF IBM                        *
*        RESTRICTED MATERIALS OF IBM                                 *
*        5685-151                                                    *
*        (C) COPYRIGHT IBM CORP. 1990, 1999                          *
*                                                                    *
*        US GOVERNMENT USERS RESTRICTED RIGHTS -                     *
*        USE, DUPLICATION OR DISCLOSURE RESTRICTED BY                *
*        GSA ADP SCHEDULE CONTRACT WITH IBM CORP.                    *
*                                                                    *
**********************************************************************
* DESCRIPTION: AOC/MVS CONFIGURATION CONTROL FILE FOR IMS CSQ        *
*              DEFINITIONS.                                          *
**********************************************************************
*   APAR#     DATE                                                   *
*  -------  --------                                                 *
*  $L1=OW35606,V1R4,17Nov98,APC(NS):New sample for CQS processing.   *
**********************************************************************
**********************************************************************
*    ACF ENTRIES FOR CQS                                             *
**********************************************************************
*
 SUBSYSTEM  IMS10C,
            DESC='CQS FOR IMS SUBSYSTEM',
            JOBTYPE=MVS,
            JOB=IMS10C,
            PARENT=(JES2),
            STRTDLY=00:05:00,
            SHUTDLY=00:01:00
*
*
* REPLIES
*
*
 IMS10C CQS0031A,
          REPLY=(,,'CONFIRM')
 IMS10C CQS0032A,
          REPLY=(,,'COLD')
 IMS10C CQS0033A,
          REPLY=(,,'COLD')
*
*
* SHUTNORM COMMANDS
*
*
 IMS10C SHUTNORM,
              CMD=(PASS1,,'MVS P &SUBSJOB'),
              CMD=(PASS2,,'MVS C &SUBSJOB')
*
*
* SHUTIMMED COMMAND
*
*
```

```
 IMS10C SHUTIMMED,
             CMD=(PASS1,,'MVS C &SUBSJOB')
*
IMS10C CQSET,
     CMD=(,,'IMSCMD &SUBSAPPL /CQSET SHUTDOWN SHAREDQ ON STRUCTURE ALL')
*
*
*THRESHOLD SETTINGS
*
*
THRESHOLDS IMS10C,
             CRIT=(02,00:30),
             FREQ=(02,01:00),
             INFR=(02,02:00)
*
*
*ABEND CODES
*
*
IMS10C IEF450I,
          CODE=(*,*,U0100,ABENDING),
          CODE=(*,*,U0104,ABENDING)
```

EVIS2CRT

```
*********************************************************************
* (C) COPYRIGHT IBM CORP. 1990, 1995 - ALL RIGHTS RESERVED          *
*********************************************************************
*                                                                   *
*        LICENSED MATERIALS - PROPERTY OF IBM                       *
*        RESTRICTED MATERIALS OF IBM                                *
*        5685-151                                                   *
*        (C) COPYRIGHT IBM CORP. 1990, 1995                         *
*                                                                   *
*        US GOVERNMENT USERS RESTRICTED RIGHTS -                    *
*        USE, DUPLICATION OR DISCLOSURE RESTRICTED BY               *
*        GSA ADP SCHEDULE CONTRACT WITH IBM CORP.                   *
*                                                                   *
*********************************************************************
*   APAR#     DATE                                                  *
*  -------  --------                                                *
*  $01=PN45452 09/09/93 MP CHANGES REQUIRED IN IMSAO FOR COMPATIBILITY *
*                          WITH AOC/MVS V1R2                        *
*                                                                   *
*********************************************************************
IMS    CRITMSGS,CODE=(*,,,SAVE)
* COPY & CREATE SDF ENTRIES FOR AOC/MVS                          @01A
SDF    IMSOLDS,PR=200,CO=R,CLEAR=(Y,RV)
SDF    IMSARCH,PR=200,CO=R,CLEAR=(Y,RV)
SDF    IMSRECN,PR=200,CO=R,CLEAR=(Y,RV)
SDF    IMSMSCL,PR=200,CO=R,CLEAR=(Y,RV)
SDF    IMSTRAN,PR=200,CO=R,CLEAR=(Y,RV)
SDF    IMSTIMR,PR=200,CO=R
SDF    CRITMSG,PR=500,CO=G
SDF    CRITMSGA,PR=500,CO=R
SDF    CRITMSGE,PR=501,CO=Y
SDF    CRITMSGW,PR=502,CO=T
SDF    CRITMSGI,PR=503,CO=G
```

EVIS2LCL

```
**********************************************************************
**********************************************************************
**                                                                  **
**       Licensed Materials - Property of IBM                       **
**       5685-151                                                   **
**       (C) Copyright IBM CORP. 1990, 1996                         **
**                                                                  **
**********************************************************************
** Name: EVIS2LCL                                                   **
** Loads RESIDENT CLISTS.                                           **
**********************************************************************
**    APAR#      Date    By Description                             **
** ----------- -------- -- ---------------------------------------- **
** $05=OW30408,V1R4,27Nov97,APC(RW): Add EVIEATHR                   **
** $04=OW20827 06/05/96 CG PTF UW90273 DID NOT INCLUDE ++HOLD ACTION *
**                         STATEMENTS INSTALLER MAY MISS NECESSARY   *
**                         INSTALLATION INFORMATION                  *
**                                                                   *
** $L1=OW19977 05/31/96 LK AOC IMS-AUTOMATION FEATURE SUPPORT FOR IMS **
**                         5.1 ARM RECOVERY                         **
**                                                                  **
** $03=OW19002 03/22/96 CG AOC/MVS IMS - FINAL TERMINATION MESSAGE  **
**                         PROCESSING IS DELAYED MSGIEF404I IEF404I **
**                         (Remove EVIET00D, Add EVIET00H)          **
**                                                                  **
** $02=OW16002 10/30/95 CG LEVEL-SET TO UPGRADE AOC/IMS R4 SERVICE  **
**                         FROM R3 SERVICE STREAM.                  **
**                                                                  **
** $01=OW06963 07/26/94 JG SAMPLES IN AEVISAMP ARE INCORRECT FOR IMS **
**                         XRF AND LCL MEMBERS.                     **
**                                                                  **
** PN17023  09/01/92   RW  IMSAO MEMBER EVIS2LCL LOADS CLISTS NO    **
**                         LONGER IN USE.                           **
**                                                                  **
** PN09862  11/11/91   NS  IMSAO SAMPLE EVIS2LCL REFERENCES CLISTS NO **
**                         LONGER USED                              **
**                                                                  **
**********************************************************************
** Function:                                                        **
**          This member specifies all the CLISTS that will be made  **
**            RESIDENT at NetView startup.                          **
**                                                                  **
**********************************************************************
RESIDENT   EVIEAAST
RESIDENT   EVIEASTS
**                                                             @05A
RESIDENT   EVIEATHR
RESIDENT   EVIEE00A
RESIDENT   EVIEE000
RESIDENT   EVIEE001
RESIDENT   EVIEI00B
RESIDENT   EVIEI00C
RESIDENT   EVIEI00D
RESIDENT   EVIEI00E
RESIDENT   EVIEI00F
RESIDENT   EVIEI00G
```

```
RESIDENT   EVIEI00I
RESIDENT   EVIEI00J
RESIDENT   EVIEI00N
**                                                                @01D
RESIDENT   EVIEI00R
RESIDENT   EVIEI003
RESIDENT   EVIEI004
RESIDENT   EVIEI005
RESIDENT   EVIEI006
RESIDENT   EVIEI007
RESIDENT   EVIEI008
RESIDENT   EVIEI009
RESIDENT   EVIEM000
RESIDENT   EVIEO00A
RESIDENT   EVIEO00B
RESIDENT   EVIEO00C
RESIDENT   EVIEO000
RESIDENT   EVIEO001
RESIDENT   EVIEO002
RESIDENT   EVIEO003
RESIDENT   EVIEO004
RESIDENT   EVIEO005
RESIDENT   EVIEO006
RESIDENT   EVIEO007
RESIDENT   EVIEO008
RESIDENT   EVIEO009
RESIDENT   EVIER000
RESIDENT   EVIER001
RESIDENT   EVIER002
RESIDENT   EVIER003
RESIDENT   EVIES002
RESIDENT   EVIES003
RESIDENT   EVIES004
RESIDENT   EVIES005
RESIDENT   EVIET00A
RESIDENT   EVIET00B
RESIDENT   EVIET00C
**         EVIET00D                                          /* @03D*/
RESIDENT   EVIET00E
RESIDENT   EVIET00H
RESIDENT   EVIET001
RESIDENT   EVIET004
RESIDENT   EVIET005
RESIDENT   EVIET006
RESIDENT   EVIET007
RESIDENT   EVIET008
RESIDENT   EVIEU00K
RESIDENT   EVIEU00P
RESIDENT   EVIEU00Q
RESIDENT   EVIEU00T
RESIDENT   EVIEU00X
RESIDENT   EVIEU007
RESIDENT   EVIEU008
RESIDENT   EVIEW000
RESIDENT   EVIEW001
RESIDENT   EVIEX002
RESIDENT   EVIEX003
RESIDENT   EVIEX004
```

```
RESIDENT   EVIEX017
RESIDENT   EVIEX100
RESIDENT   EVIEX102
```

EVIPRFAO

```
***********************************************************************
*  (C) COPYRIGHT IBM CORP. 1990, 1995 - ALL RIGHTS RESERVED          *
***********************************************************************
*                                                                     *
*        LICENSED MATERIALS - PROPERTY OF IBM                         *
*        RESTRICTED MATERIALS OF IBM                                  *
*        5685-151                                                     *
*        (C) COPYRIGHT IBM CORP. 1990, 1995                           *
*                                                                     *
*        US GOVERNMENT USERS RESTRICTED RIGHTS -                      *
*        USE, DUPLICATION OR DISCLOSURE RESTRICTED BY                 *
*        GSA ADP SCHEDULE CONTRACT WITH IBM CORP.                     *
*                                                                     *
***********************************************************************
*   APAR#     DATE                                                    *
*  -------  --------                                                  *
*  $01=OW07849 09/12/94 JS AOC IMS INITIALIZATION PROBLEMS AFTER AOC   *
*                          APAR OW03453                               *
*                                                                     *
***********************************************************************
* DESCRIPTION    :  PROFILE FOR THE AUTOMATED OPERATOR                *
***********************************************************************
EVIPRFAO     PROFILE IC=EVIEIAAI
             AUTH     CTL=GLOBAL
             OPCLASS  1,2
             END
```

Appendix C. DB2 in an XRF Environment

This section presents the major results of DB2 running with XRF IMS. Our general conclusions and recommendations are as follows:

- In a multi-CEC environment, ensure that DB2 has been stopped on the active subsystem before starting it on the alternate subsystem. Using the same DB2 subsystem from both CECs probably will lead to inconsistent data. One bypass can be to modify the DSNMSTR procedure, by adding a specific data set with a disposition of "OLD." This bypass has been documented in an ITSC-ST flash.
- On the alternate subsystem, do not start a DB2-oriented Message Processing Region either before the takeover takes place or before DB2 is started on the alternate.
- -923 SQL code:

  `CONNECTION NOT ESTABLISHED, ......`

  should be handled by the application. This can be done by specifying the "R" error option on the SSM user entry. In this case, DB2 probably needs to be started with a `-START DB2` command.
- The "-924" SQL error code should be handled by the application program (see the "R" error option on the SSM user entry). In this case, the connection between DB2 and IMS probably needs to be started with a `/START SUBSYS DSN` command.

Note: The following section contains a few blank pages. This has been done to keep illustrations and their descriptions on facing pages, for the reader's convenience.

Takeover Overview—Example 1

The purpose of this first example is to present a general scenario using two MVS systems running with IMS-XRF. DB2 is part of the overall process, and the events are presented in chronological order so that the reader can see the flow of what is going on with both active and alternate subsystems.

We do not show all messages appearing on the MVS system log, but only those we consider crucial to understanding the sequence and the relationship between the events.

Conclusions of Example 1

Following a strict order of the operations on both subsystems allows you to control the takeover, even when DB2 is involved with the processing. The main recommendation is to make sure that DB2 has been stopped on the active before starting it on the alternate subsystem. This should be verified before switching from the active to the alternate.

Takeover Overview—Example 1 (contd.)

1 On the active subsystem, issue `-START DB2` from the MVS console.

2 Also start IMS/VS. The order of 1 and 2 does not matter.

3 Start IMS on the alternate subsystem.

4 From the alternate IMS Master Terminal (or from the MVS console), tell IMS that System 2 is the alternate.

5 On the active subsystem, start the VTAM ACB and one Message Processing Region.

6 From the alternate, start the VTAM ACB and one Message Processing Region.

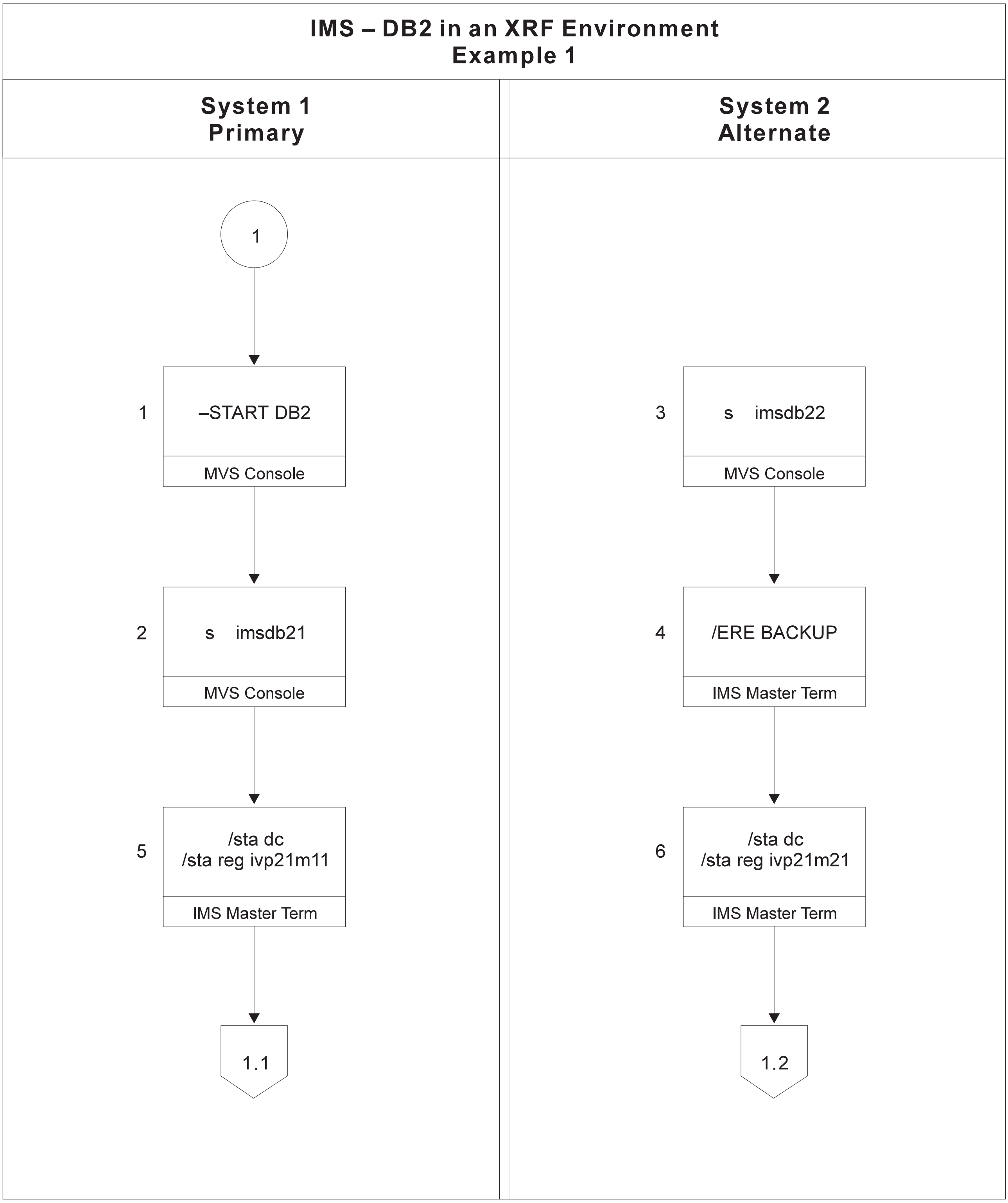

Figure 20. DB2/XRF Takeover Overview

Takeover Overview—Example 1 (contd.)

7 On the active subsystem from an IMS terminal, issue the IMS transaction "DSN8CS," which is one of the sample IMS-DB2 transactions.

8 If we issue a /DIS A command from the alternate subsystem at this time, we get information about the status of IMS regions.

9 This is a subset of DFS000I messages from the /DIS A command:

```
DFS000I     REGID  JOBNAME   TYPE   TRAN/STEP PROGRAM
DFS000I         1  MSGRGN21  TP     WAITING FOR SWITCHOVER
DFS000I            DBRCJBB2  DBRC   WAITING FOR SWITCHOVER
DFS000I     VTAM ACB OPEN           -LOGONS DISABLED
```

10 From the active, we issue an MVS CANCEL command against the IMS Control Region, just before a takeover. This will simulate an abnormal condition which could occur on the active.

11 Among the messages received on the system console, we can see:

```
 AVM004I TAKEOVER IN PROGRESS FOR SUBSYSTEM IMSA, ACTIVE ELEMENT OF RSE
  IMSXRF

*AVM011E ENSURE A TAKEOVER IS IN PROGRESS FOR THE BACKUP ELEMENT OF RSE
  IMSXRF

*AVM006E TELL OPERATOR AT BACKUP TO REPLY "GO" TO MESSAGE AVM005A.
  I/O PREVENTION IS COMPLETE FOR SUBSYSTEM IMSA, FAILING ACTIVE ELEMENT
  OF RSE   IMSXRF
```

12 Issue the -STOP DB2 command from the active MVS console. You must ensure that DB2 cannot be started on the alternate until it has stopped on the active. This is because DB2 does not provide data sharing capability or control. Failing to stop DB2 on the active will cause inconsistent data.

From this point on, the active subsystem is no longer involved with XRF operations, except that it can become the "new" alternate after the takeover from System 1 to System 2 completes.

13 Now we tell System 2 to become the new active. The IMS /SWI SYSTEM FORCE command is issued for this purpose. We will then receive messages summarized below.

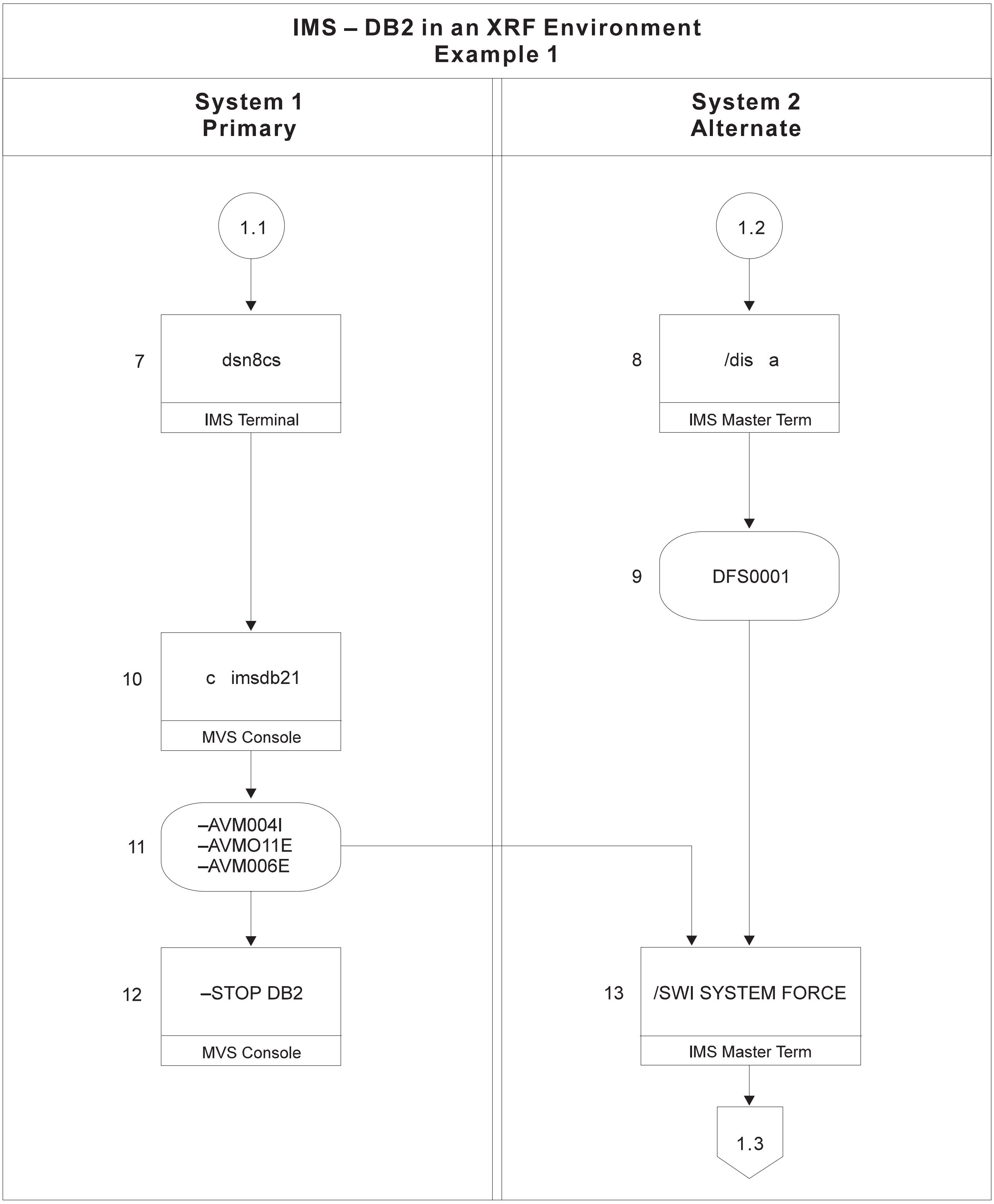

Figure 21. DB2/XRF Takeover Overview (contd.)

Takeover Overview—Example 1 (contd.)

14 At switch time, four messages are noteworthy:

```
*DFS3890I TAKEOVER REQUESTED REASON CODE = 7A.   IMSB
*DFS3891I TAKEOVER IN PROGRESS IMSB
*DFS994I TAKEOVER COMPLETED. IMSB
*DFS3860I hh:mm:ss ALL TERMINAL SESSIONS SWITCHED. IMSB
```

15 We now have to reply "GO" on the alternate subsystem when I/O prevention has been completed (see message "AVM006E" above in Note 11). The message is:

```
*xx AVM005A REPLY GO WHEN I/O PREVENTION COMPLETES FOR RSE IMSXRF
```

As a result of replying "GO," we get the following message on the MVS console:

```
DFS0488I - UNLOCK COMMAND COMPLETED. RC = 00 IMSB
```

16 **Now System 2 is the new active.**

17 We now have to `-START DB2` on the new active.

18 A `/DIS SUBSYS ALL` command will give us the following messages:

```
DFS000I    SUBSYS    CRC REGID PROGRAM  LTERM    STATUS    IMSB
DFS000I    DSN       -                           CONN      IMSB
```

19 From our IMS terminal, we now get the message:

```
DFS3904 hh:mm:ss SYSTEM TAKEOVER OCCURRED.
RE-ENTER THE LAST INPUT.
```

We are now processing IMS transactions on the alternate (now the active).

Note:

Comparing this example with example 3:

- There are no URs to be completed after the takeover. There are only new URs.
- SQL code -924 is not returned, whereas it is returned if URs need to be completed after the takeover.

  At this time, we cannot explain this behavior.

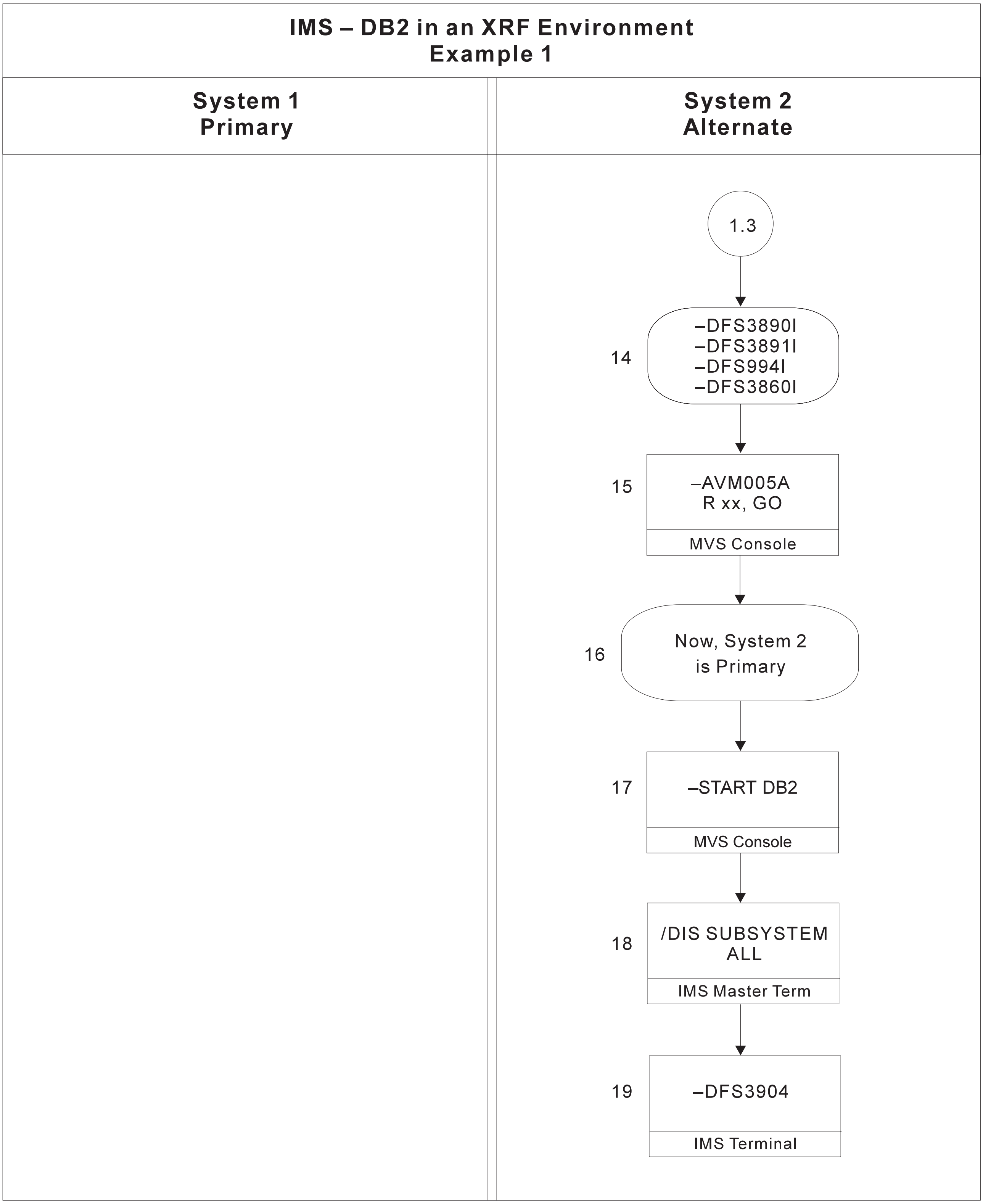

Figure 22. DB2/XRF Takeover Overview (contd.)

Normal Takeover Sequence—Example 2

This example shows the recommended sequence of events on both subsystems in order to make a takeover as transparent as possible to the end user. The end user will notice nothing unusual except a longer response time.

To simulate a real situation, the takeover has been done after having performed a "System Reset" on the active subsystem, thus causing IMS and DB2 to stop abnormally. We also modified the sample transaction "DSN8CS" by issuing a WTOR to the MVS console. This leaves DSN8CS "in-flight" across the takeover.

Conclusions of Example 2

If we have an in-flight, DB2-oriented transaction when the active subsystem fails, the transaction will automatically be re-queued and re-executed on the alternate subsystem after takeover, provided that:

- There is no prestarted MPP on the alternate subsystem, and
- DB2 is started on the alternate before the takeover takes place.

Normal Takeover Sequence—Example 2

1 Start IMS from the active subsystem MVS console.

2 From the active subsystem IMS Master Terminal:

- Start communications (the VTAM ACB)
- Start the Message Processing Region
- Display the status of all subsystems.

The status of DB2 is the following:

```
DFS000I     SUBSYS    CRC   REGID  PROGRAM  LTERM  STATUS   IMSA
DFS000I     DSN       -                            NOT CONN IMSA
```

3 Start IMS from the alternate subsystem MVS console.

4 From the alternate subsystem IMS Master Terminal, issue `/ERE BACKUP` to define System 2 as the alternate IMS subsystem.

5 From the alternate subsystem IMS Master Terminal:

- Start the VTAM ACB
- Display what is active in IMS.

You receive the following from the `/DIS A` command:

```
DFS000I     REGID  JOBNAME   TYPE   TRAN/STEP  PROGRAM CLASS   IMSB
DFS000I            MSGRGN    TP     NONE                       IMSB
DFS000I            BATCHREG  BMP    NONE                       IMSB
DFS000I            FPRGN     FP     NONE                       IMSB
DFS000I            DBRCJBB2  DBRC   WAITING FOR SWITCHOVER     IMSB
DFS000I            DLIJBB2   DLS                               IMSB
DFS000I     VTAM ACB OPEN           -LOGONS DISABLED           IMSB
```

6 `-START DB2` from the active subsystem MVS console.

7 From the active subsystem IMS Master Terminal, display the status of all subsystems. The status of DB2 is as follows:

```
DFS000I     SUBSYS    CRC   REGID  PROGRAM  LTERM  STATUS   IMSA
DFS000I     DSN       -                            CONN     IMSA
```

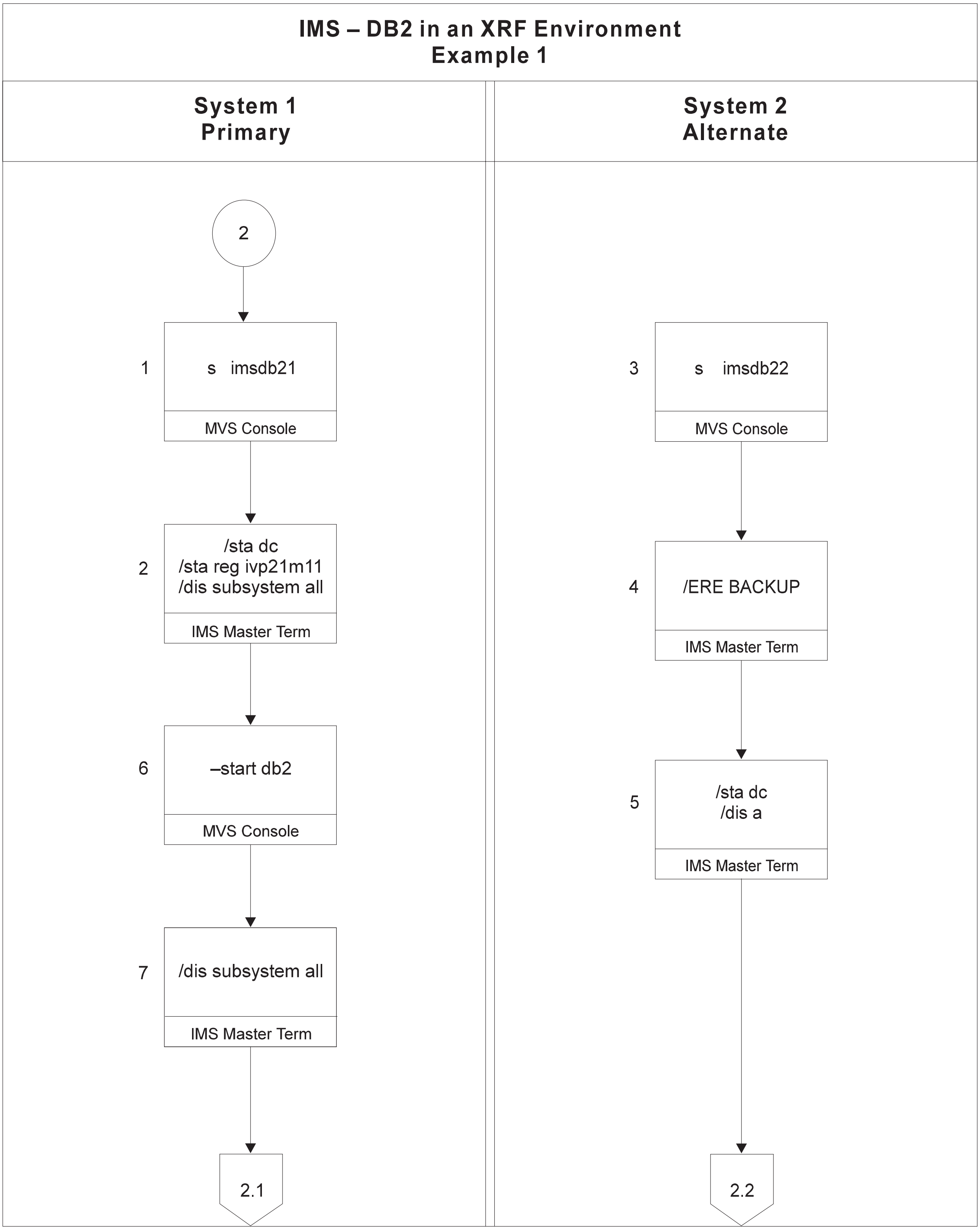

Figure 23. DB2/XRF Normal Takeover Sequence

Normal Takeover Sequence—Example 2 (contd.)

8 From one active subsystem IMS terminal, issue the sample transaction "DSN8CS." (Note that this transaction has been modified in order to keep it in-flight during the takeover tests.)

9 Issuing -DISPLAY THREAD(*) from the active subsystem MVS console gives the following:

```
DSNV401I - DISPLAY THREAD REPORT FOLLOWS -
DSNV402I - ACTIVE THREADS
NAME     ST A   REQ ID           AUTHID   PLAN      ASID
IMSXRF   T     1362 0001DSN8IC02 MTO      DSN8IC02   0010
IMSXRF   N        2              IBMUSER             01F6
DISPLAY ACTIVE REPORT COMPLETE
```

10 In order to force the takeover, we "System Reset" and re-IPL the active subsystem.

11 From the System 2 (Alternate) MVS console, issue -START DB2. As System 1 has been re-IPLed, we know that DB2 has been abnormally terminated.

During DB2 startup, the following major messages occurred:

```
DSNY001I - SUBSYSTEM STARTING
IEC161I 056-084,DSNMSTR,DSNMSTR,BSDS1,,,DSNC120.BSDS01,
IEC161I DSNC120.BSDS01.DATA,CATALOG.IXRFDB2
        (these 2 lines are repeated for data and index of both
         BSDS copies. They are the well-known VSAM verify messages
         issued at OPEN time when a previous CLOSE has not been
         successful.)
DSNR001I - RESTART INITIATED
DSNR003I - RESTART...PRIOR CHECKPOINT RBA=000000E57372
DSNR004I - RESTART...UR STATUS COUNTS
IN COMMIT=1, INDOUBT=0, INFLIGHT=1, IN ABORT=0
DSNR007I - RESTART...STATUS TABLE
T  CON-ID    CORR-ID     AUTHID    PLAN   S     URID     DAY   TIME
- -------- ------------ -------- -------- - ------------ --- --------
I DSN      DSN          SYSOPR            C 000000E67DB0 274 22:04:18
S IMSXRF   0001DSN8IC02 MTO      DSN8IC02 F 000000E67C09 274 22:04:18
DSNR005I - RESTART...COUNTS AFTER FORWARD RECOVERY
IN COMMIT=0, INDOUBT=0
DSNR006I - RESTART...COUNTS AFTER BACKWARD RECOVERY
INFLIGHT=0, IN ABORT=0
DSNR002I - RESTART COMPLETED
DSN9022I - DSNYASCP -START DB2 NORMAL COMPLETION
```

12 A -DISPLAY THREAD(*) command issued from the alternate system MVS console gives:

```
DSNV419I - NO ACTIVE CONNECTIONS FOUND
```

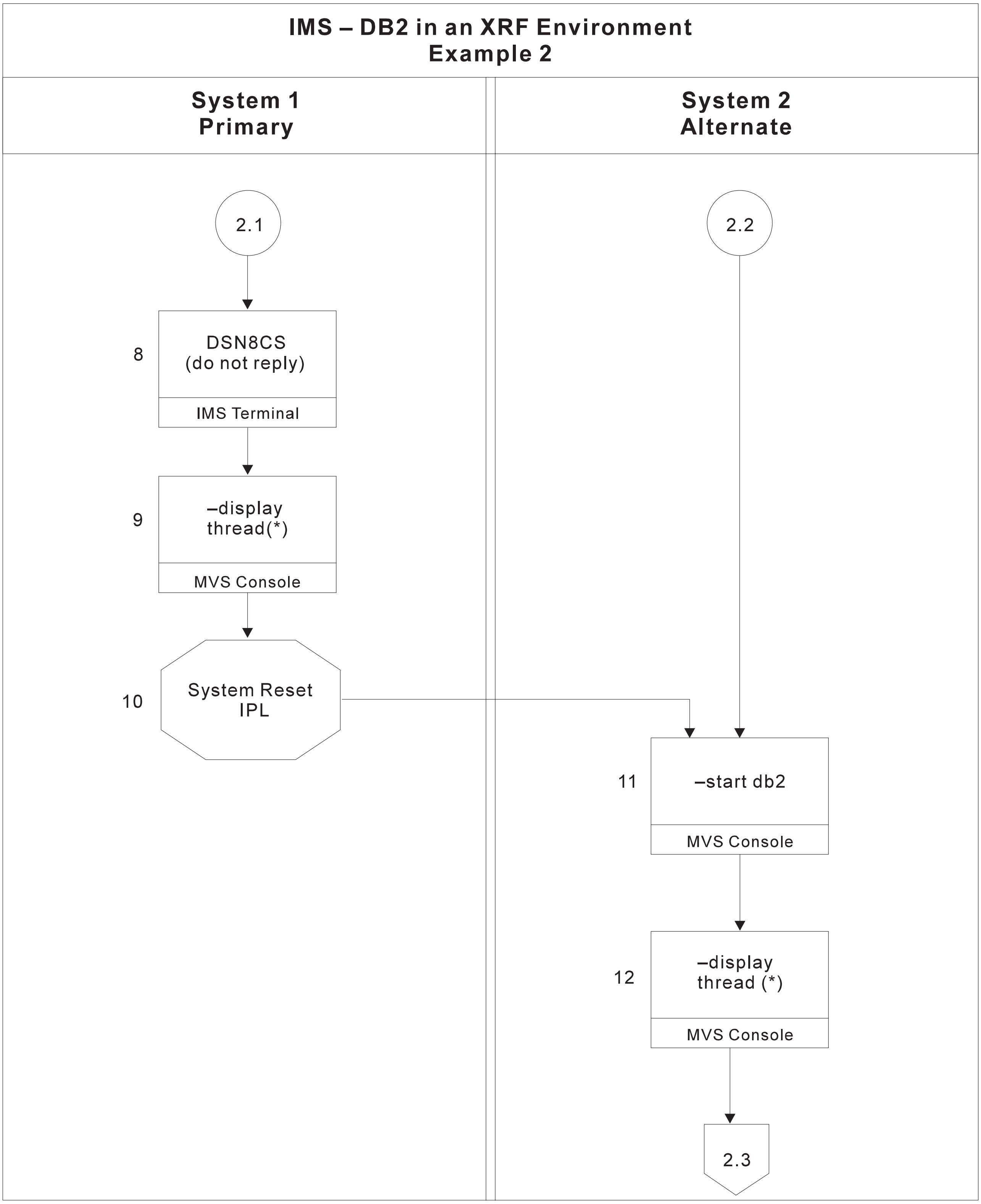

Figure 24. DB2/XRF Normal Takeover Sequence (contd.)

Normal Takeover Sequence—Example 2 (contd.)

13 From the alternate subsystem IMS Master Terminal, issuing the "/SWI SYSTEM FORCE" command will force the takeover.

14 We will then get the following major messages:

```
*DFS3890I TAKEOVER REQUESTED REASON CODE = 78
*DFS3891I TAKEOVER IN PROGRESS
DFS994I TAKEOVER COMPLETED
DFS3860I 22:16:41 ALL TERMINAL SESSIONS SWITCHED. IMSB
```

15 Issuing the `/DIS SUBSYS ALL` command from the IMS Master Terminal will give:

```
DFS000I     SUBSYS    CRC   REGID  PROGRAM  LTERM  STATUS   IMSB
DFS000I     DSN       -                            CONN     IMSB
```

16 We can now reply "GO" to the message:

```
AVM005A REPLY GO WHEN I/O PREVENTION COMPLETES FOR RSE IMSXRF
```

As a result, we will get the messages:

```
DFS0488I - UNLOCK COMMAND COMPLETED. RC = 00  IMSB
DFSAOUE0: DFS0488I - UNLOCK COMMAND COMP
```

From now on, System 2 is the active.

17 Start a Message Processing Region.

The "DSN8CS" transaction is re-queued and re-executed automatically.

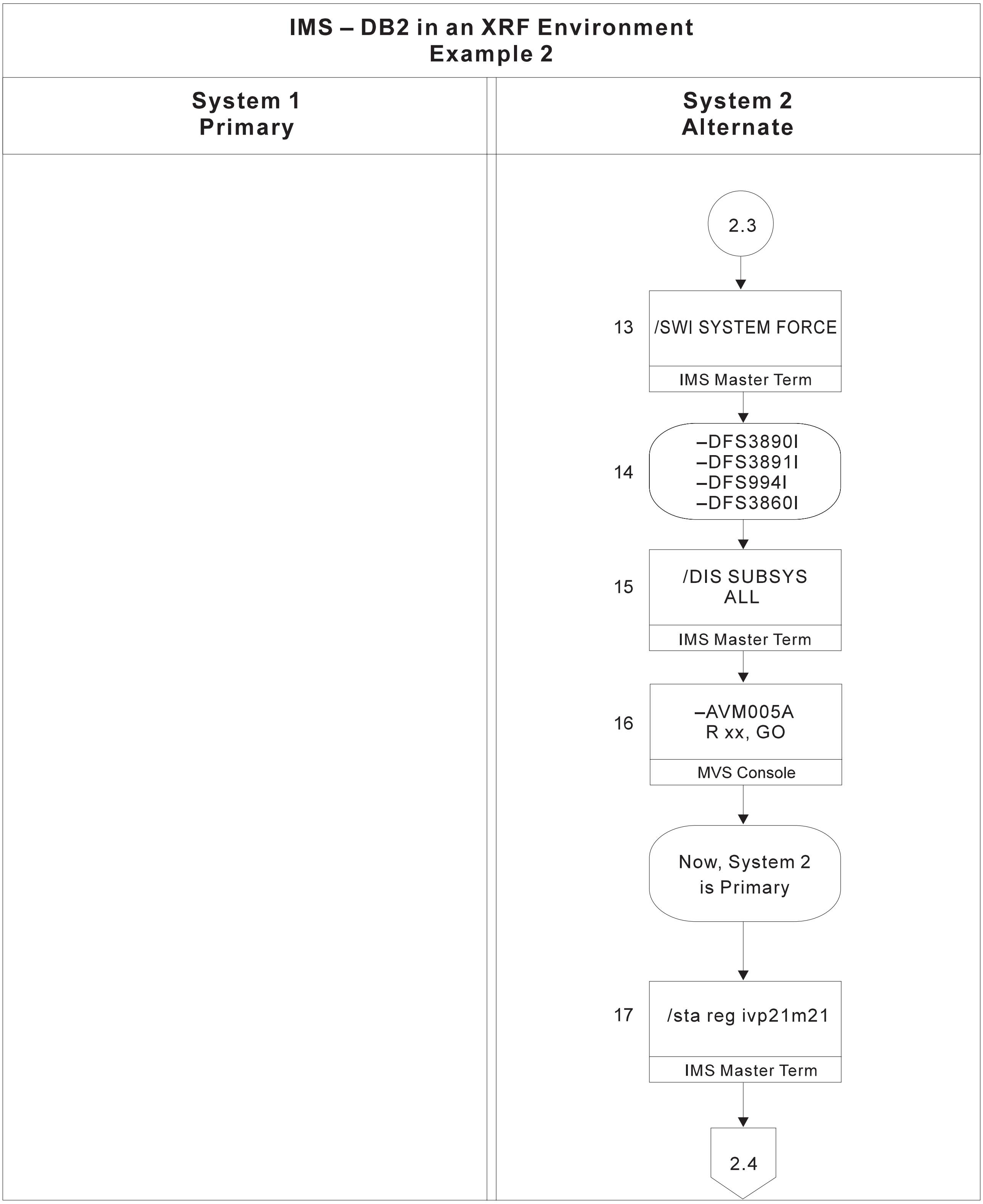

Figure 25. DB2/XRF Normal Takeover Sequence (contd.)

Normal Takeover Sequence—Example 2 (contd.)

18 The WTOR message for “DSN8CS” is issued to the MVS console as the transaction executes.

19 Issuing -`DISPLAY THREAD(*)` DB2 command gives the following:

```
DSNV401I - DISPLAY THREAD REPORT FOLLOWS -
DSNV402I - ACTIVE THREADS
NAME     ST A   REQ ID           AUTHID   PLAN       ASID
IMSXRF   T        4 0001DSN8IC02 MTO      DSN8IC02   0013
IMSXRF   N      124              IBMUSER             0015
DISPLAY ACTIVE REPORT COMPLETE
```

20 Now reply to the outstanding WTOR message to allow the DSN8CS transaction to continue.

21 Transaction “DSN8CS” just resumes with the last panel.

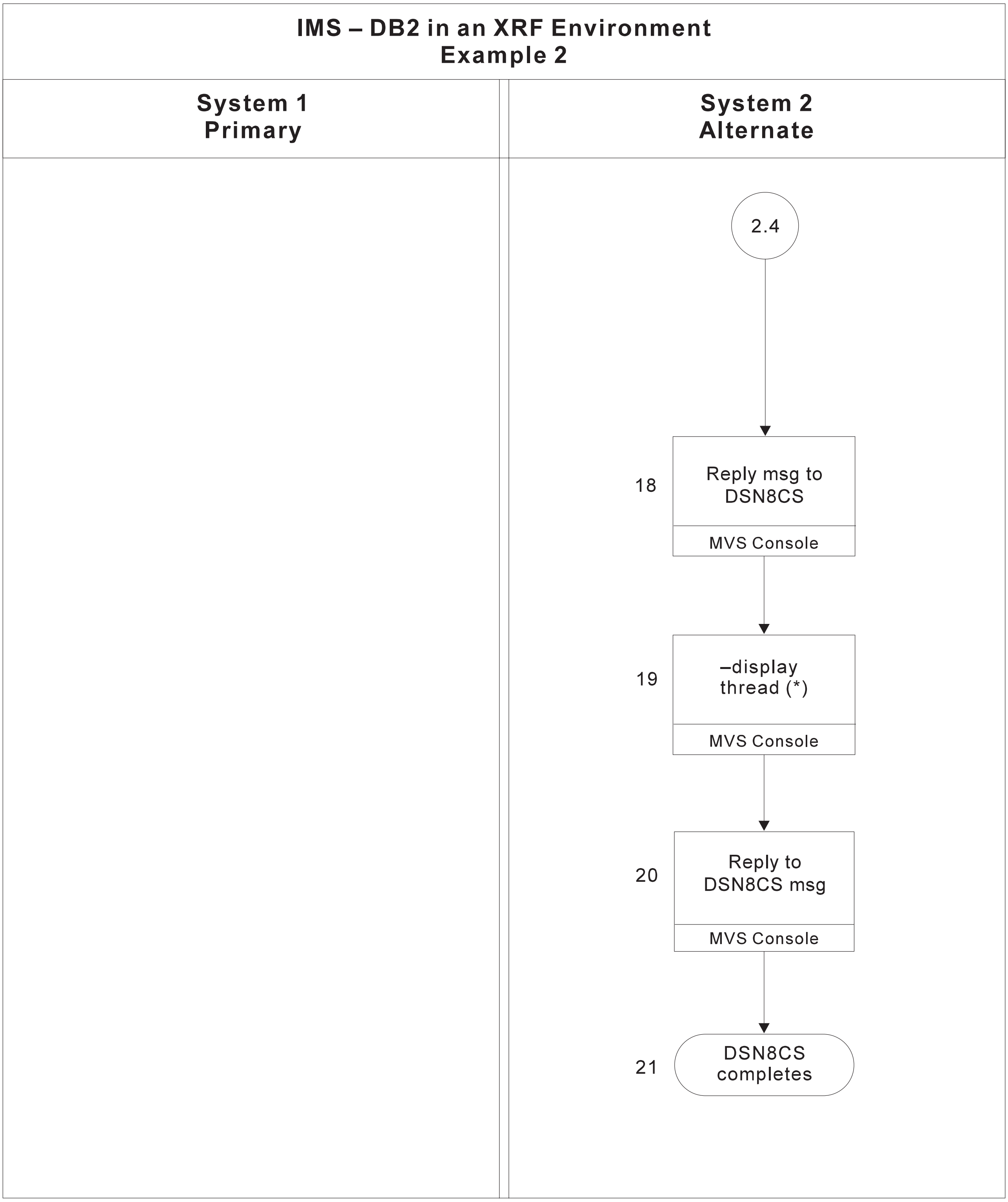

Figure 26. DB2/XRF Normal Takeover Sequence (contd.)

Forced Takeover, Cancel IRLM/IMS—Example 3

In this example, the takeover has been forced by cancelling:

- The IRLM procedure, which forced DB2 to abnormally terminate
- Then, IMS

We used the modified sample transaction “DSN8CS” so that it could be in-flight across the takeover. We did that by issuing a WTOR to the MVS console.

Conclusions of Example 3

If we have an in-flight, DB2-oriented transaction when the active subsystem fails, the in-flight transaction will be automatically backed out and will return an -923 SQL code if the following sequence is used:

- An MPP has been started on the alternate subsystem **before** the takeover.
- DB2 has been started on the alternate **after** the takeover.

Forced Takeover, Cancel IRLM/IMS—Example 3 (contd.)

1 Start IMS from the active subsystem MVS console.

2 From the active subsystem IMS Master Terminal:

- Start the VTAM ACB
- Start the Message Processing Region.

3 `-START DB2` from the active subsystem MVS console.

4 From the active subsystem IMS Master Terminal, display the active IMS tasks. The status is as follows:

```
DFS000I     REGID  JOBNAME   TYPE   TRAN/STEP  PROGRAM CLASS   IMSA
DFS000I            MSGRGN11  TP     WAITING                    IMSA
DFS000I            BATCHREG  BMP    NONE                       IMSA
DFS000I            FPRGN     FP     NONE                       IMSA
DFS000I            DBRCJBB1  DBRC                              IMSA
DFS000I            DLIJBB1   DLS                               IMSA
DFS000I     VTAM ACB OPEN           -LOGONS ENABLED            IMSA
DFS000I     *86275/233507*   IMSXRF     ACTIVE                 IMSA
```

5 From the active subsystem IMS Master Terminal, display the status of all subsystems related to IMS. The status of DB2 is:

```
DFS000I     SUBSYS    CRC   REGID  PROGRAM  LTERM  STATUS    IMSA
DFS000I     DSN       -                            CONN      IMSA
```

6 Start IMS from the alternate subsystem MVS console.

7 From the alternate subsystem IMS Master Terminal, issue:

```
/ERE BACKUP
```

in order to define System 2 as the alternate IMS subsystem.

8 From the alternate subsystem MVS console, start the VTAM ACB.

9 From the alternate subsystem IMS Master Terminal, start the IMS Message Processing Region.

10 From the alternate subsystem IMS Master Terminal, display what is active for the alternate IMS subsystem.

```
DFS000I     REGID  JOBNAME   TYPE   TRAN/STEP  PROGRAM CLASS   IMSB
DFS000I            MSGRGN21  TP     WAITING FOR SWITCHOVER     IMSB
DFS000I            BATCHREG  BMP    NONE                       IMSB
DFS000I            FPRGN     FP     NONE                       IMSB
DFS000I            DBRCJBB2  DBRC   WAITING FOR SWITCHOVER     IMSB
DFS000I            DLIJBB2   DLS                               IMSB
DFS000I     VTAM ACB OPEN           -LOGONS DISABLED           IMSB
DFS000I     *86275/233721*   IMSXRF    BACKUP - TRACKING       IMSB
```

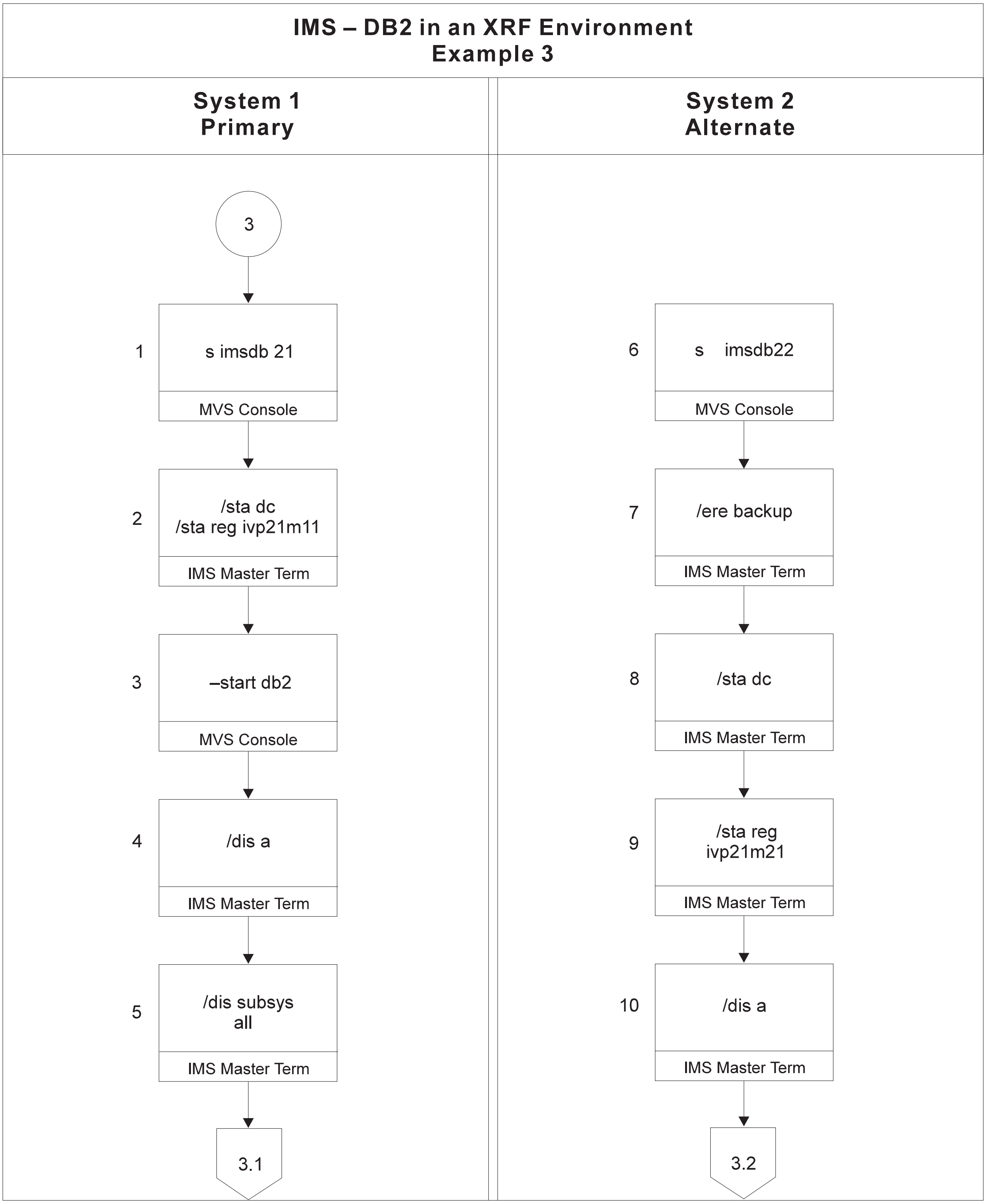

Figure 27. DB2/XRF Forced Takeover, Cancel IRLM/IMS

Forced Takeover, Cancel IRLM/IMS—Example 3 (contd.)

11 From one active subsystem IMS terminal, issue the sample transaction "DSN8CS." (Note that this transaction has been modified in order to keep it in-flight during the takeover tests.)

12 Issuing -`DISPLAY THREAD(*)` from the active subsystem MVS console gives the following:

```
DSNV401I - DISPLAY THREAD REPORT FOLLOWS -
DSNV402I - ACTIVE THREADS
NAME     ST A   REQ ID           AUTHID   PLAN       ASID
IMSXRF   T       47 0001DSN8IC02 MTO      DSN8IC02   0013
IMSXRF   N        2              IBMUSER             0029
DISPLAY ACTIVE REPORT COMPLETE
```

13 Cancel the DB2 IRLM procedure to simulate a failure for System 1.

14 In order to force the takeover, cancel IMS.

15 From the alternate subsystem IMS Master Terminal, issuing the `/SWI SYSTEM FORCE` command will force the takeover.

16 We then get the following major messages:

```
*DFS3890I TAKEOVER REQUESTED REASON CODE = 78
*DFS3891I TAKEOVER IN PROGRESS
DFS994I TAKEOVER COMPLETED
DFS3860I 23:41:05 ALL TERMINAL SESSIONS SWITCHED. IMSB
```

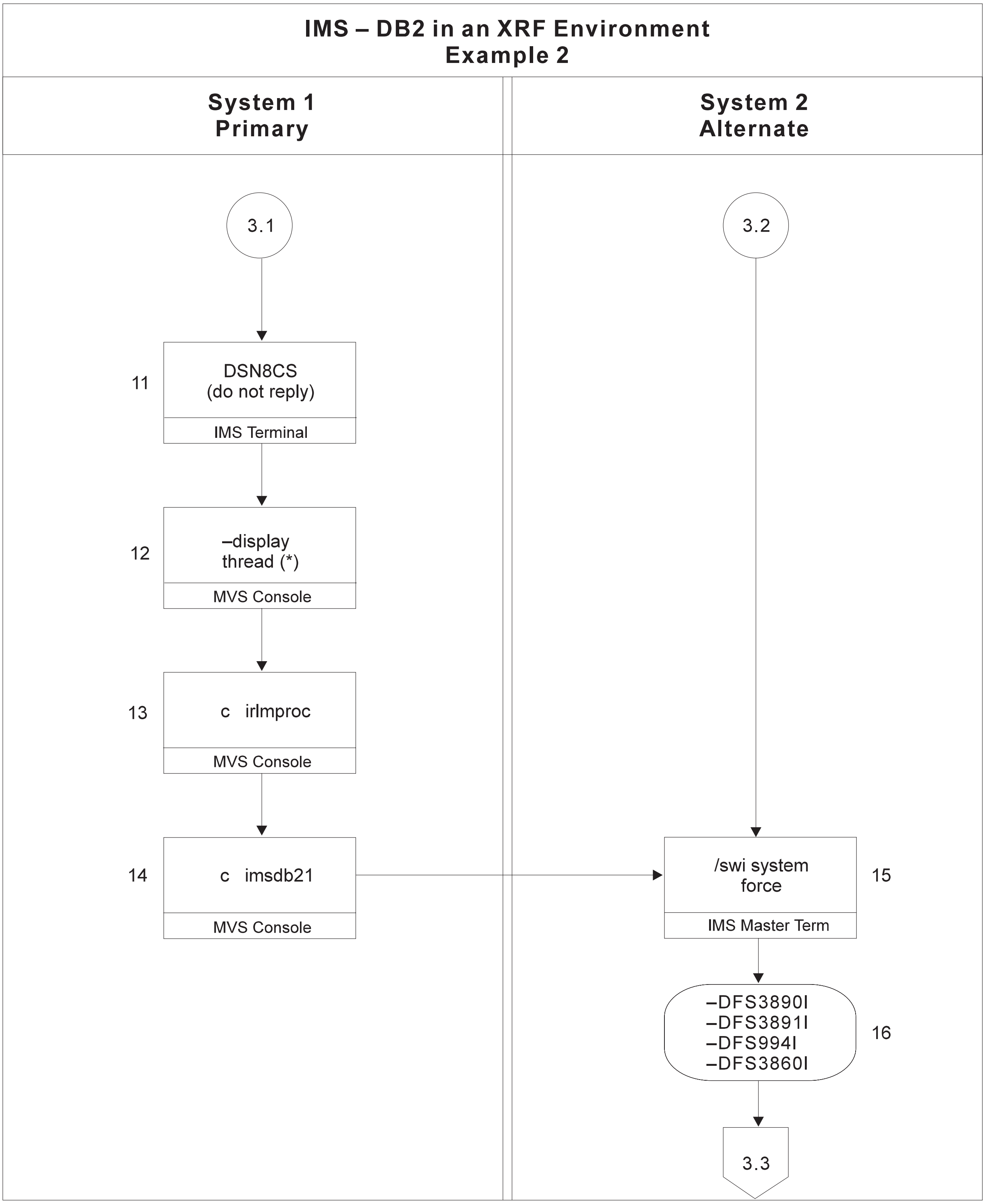

Figure 28. DB2/XRF Forced Takeover, Cancel IRLM/IMS (contd.)

Forced Takeover, Cancel IRLM/IMS—Example 3 (contd.)

17 We can now reply "GO" to the message:

```
AVM005A REPLY GO WHEN I/O PREVENTION COMPLETES FOR RSE IMSXRF
```

As a result, we will get the messages:

```
DFS0488I - UNLOCK COMMAND COMPLETED. RC = 00  IMSB
DFSAOUE0: DFS0488I - UNLOCK COMMAND COMP
```

From now on, System 2 is the new active.

18 Issue the `/DIS A` IMS command:

```
DFS000I     REGID  JOBNAME   TYPE   TRAN/STEP  PROGRAM CLASS   IMSB
DFS000I            MSGRGN21  TP     WAITING                    IMSB
DFS000I            BATCHREG  BMP    NONE                       IMSB
DFS000I            FPRGN     FP     NONE                       IMSB
DFS000I            DBRCJBB2  DBRC                              IMSB
DFS000I            DLIJBB2   DLS                               IMSB
DFS000I     VTAM ACB OPEN           -LOGONS ENABLED            IMSB
DFS000I     *86275/234120*   IMSXRF    ACTIVE                  IMSB
```

19 From System 2 (alternate) MVS console, issue `-START DB2`.

During DB2 startup, the main following messages occurred:

```
DSNY001I - SUBSYSTEM STARTING
IEC161I 056-084,DSNMSTR,DSNMSTR,BSDS1,,,DSNC120.bsds01,
IEC161I DSNC120.BSDS01.DATA,CATALOG.IXRFDB2
        (these 2 lines are repeated for data and index of both
         BSDS copies. They are the well known VSAM verify messages
         issued at OPEN time when a previous CLOSE has not been
         successful.)
DSNR001I - RESTART INITIATED
DSNR003I - RESTART...PRIOR CHECKPOINT RBA=000000F38060
DSNR004I - RESTART...UR STATUS COUNTS
IN COMMIT=0, INDOUBT=0, INFLIGHT=1, IN ABORT=0
DSNR007I - RESTART...STATUS TABLE
T  CON-ID    CORR-ID     AUTHID    PLAN   S     URID     DAY   TIME
- -------- ------------ -------- -------- - ------------ --- --------
S IMSXRF   0001DSN8IC02 MTO      DSN8IC02 F 000000F3C807 275 23:38:40
DSNR005I - RESTART...COUNTS AFTER FORWARD RECOVERY
IN COMMIT=0, INDOUBT=0
DSNR006I - RESTART...COUNTS AFTER BACKWARD RECOVERY
INFLIGHT=0, IN ABORT=0
DSNR002I - RESTART COMPLETED
DSN9022I - DSNYASCP -START DB2 NORMAL COMPLETION
```

20 A `-DISPLAY THREAD(*)` command issued from the alternate System MVS console gives:

```
DSNV419I - NO ACTIVE CONNECTIONS FOUND
```

21 Issuing the `/DIS SUBSYS ALL` command from the IMS Master Terminal will give:

```
DFS000I     SUBSYS    CRC   REGID  PROGRAM  LTERM  STATUS   IMSB
DFS000I     DSN       -                            STOPPED  IMSB
```

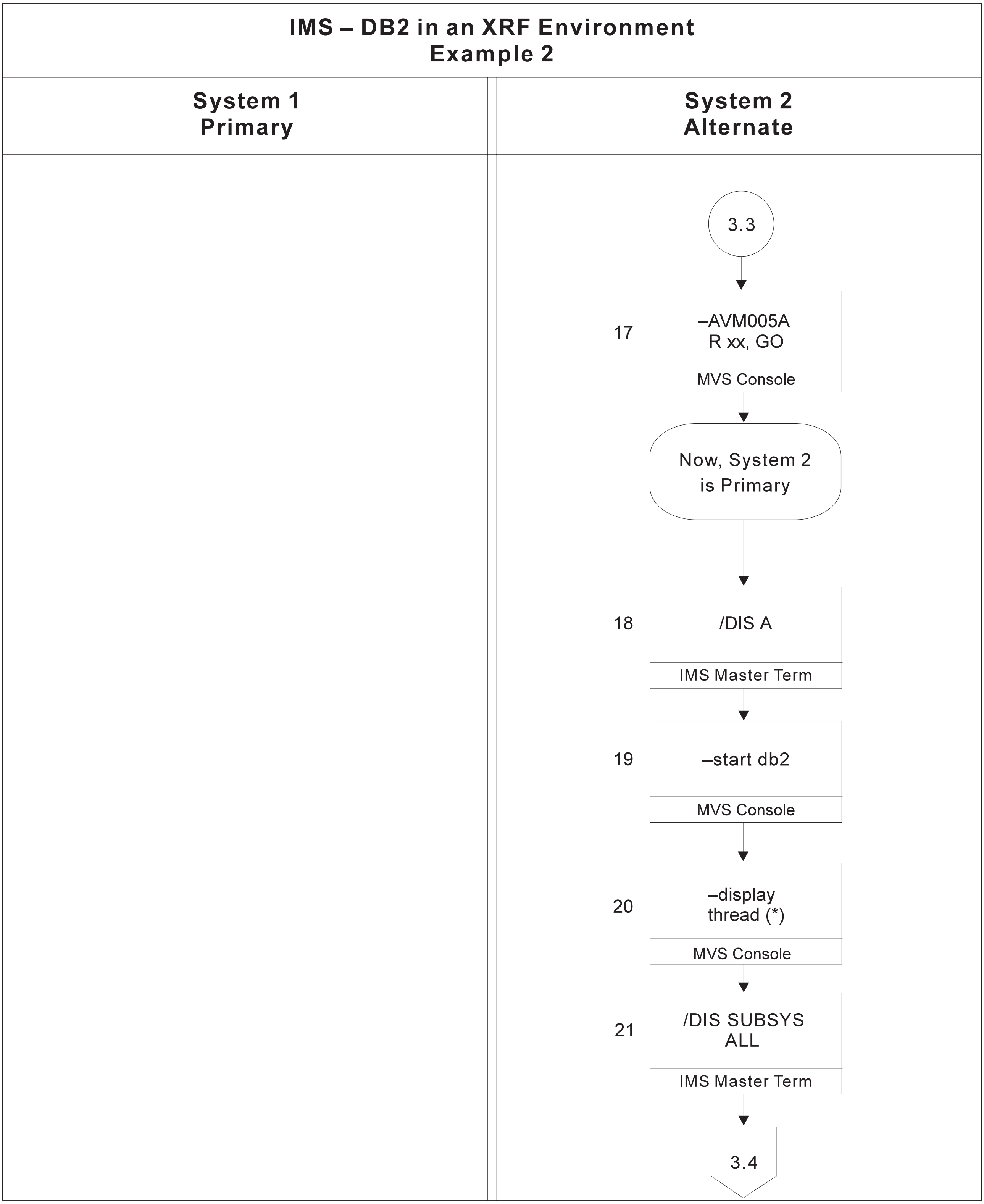

Figure 29. DB2/XRF Forced Takeover, Cancel IRLM/IMS (contd.)

Forced Takeover, Cancel IRLM/IMS—Example 3 (contd.)

22 The "DSN8CS" transaction is backed out and re-executed automatically, giving a -923 SQL code because DB2 has a "stopped" status.

In fact, the transaction has been backed out and re-executed as soon as System 2 has become the active (see Figure 29 on page 352, box 17). It has only been seen by the user at this stage, because we had to have the terminal connected to IMS first.

23 Issue `/START SUBSYS DSN` command.

24 The status of DB2 is now as follows:

```
DFS000I     SUBSYS    CRC   REGID  PROGRAM  LTERM  STATUS   IMSB
DFS000I     DSN       -                            CONN     IMSB
```

25 The first panel of "DSN8CS" transaction is displayed. This is part of the application's design.

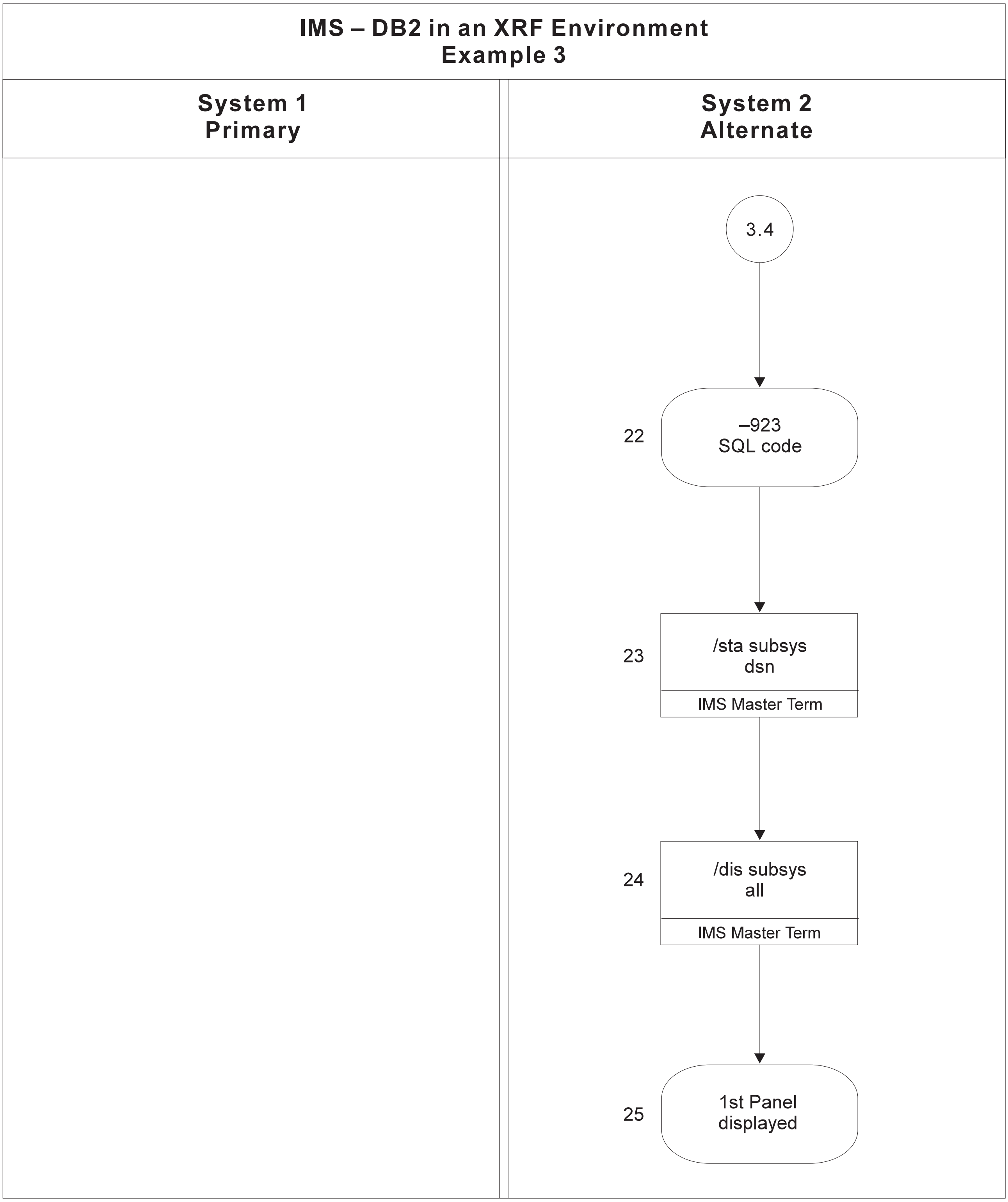

Figure 30. DB2/XRF Forced Takeover, Cancel IRLM/IMS (contd.)

Forced Takeover, Cancel IMS/IRLM—Example 4

In this example, the takeover has been forced by cancelling:

- IMS, then
- The IRLM procedure, which forced DB2 to abnormally terminate.

We used the modified sample transaction “DSN8CS” so that it could be in-flight across the takeover. We did that by issuing a WTOR to the MVS console.

Conclusions of Example 4

If we have an in-flight, DB2-oriented transaction when the active subsystem fails, the in-flight transaction will be automatically backed out and will return an -924 SQL code, if the following sequence is used:

- An MPP has been started on the alternate subsystem **before** the takeover.
- DB2 has been started on the alternate **before** the takeover took place.

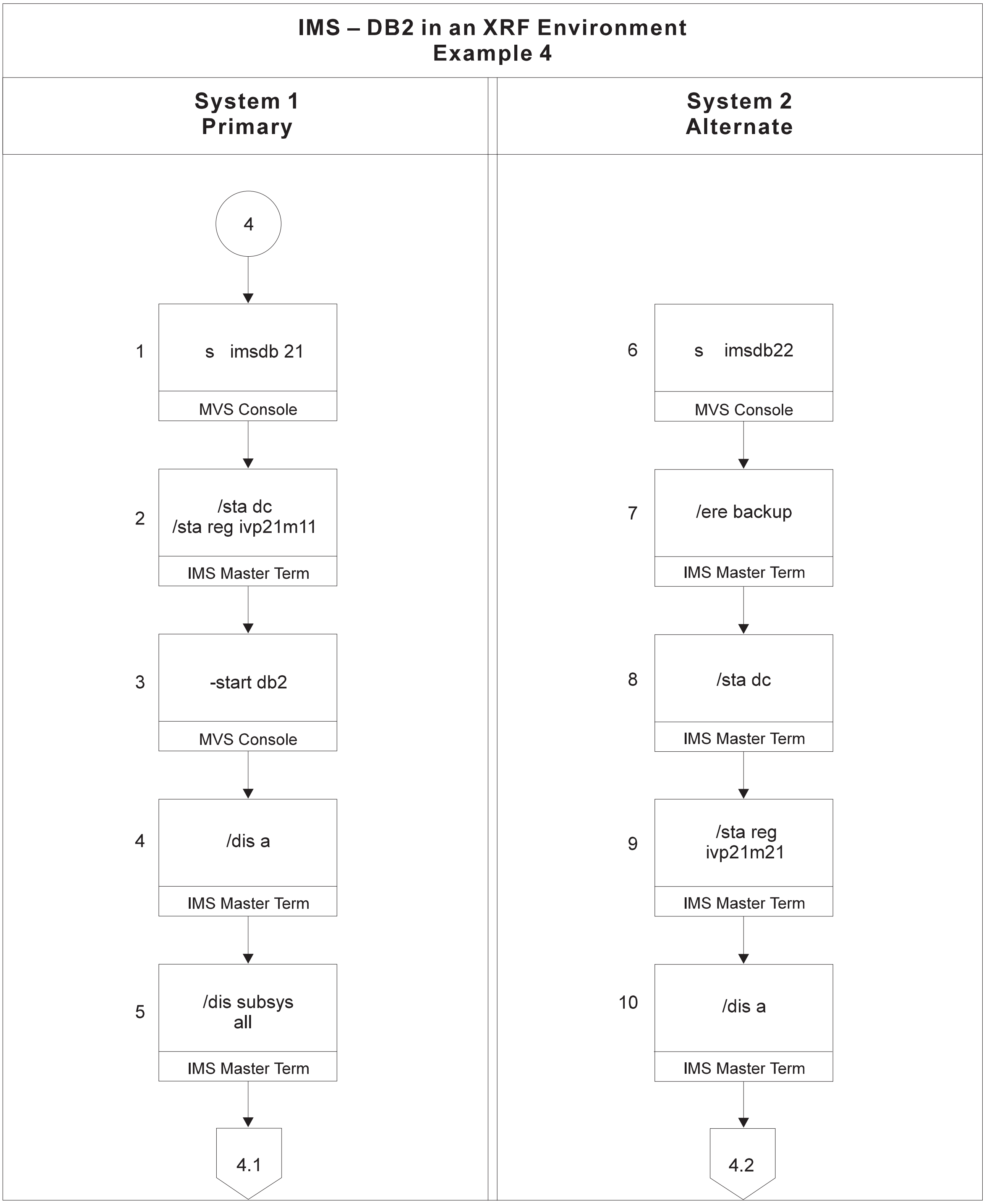

Figure 31. DB2/XRF Forced Takeover, Cancel IMS/IRLM

Forced Takeover, Cancel IMS/IRLM (contd.)

1 Start IMS from the active subsystem MVS console.

2 From the active subsystem IMS Master Terminal:

- Start the VTAM ACB
- Start the Message Processing Region.

3 `-START DB2` from the active subsystem MVS console.

4 From the active subsystem IMS Master Terminal, display active IMS tasks. The status is as follows:

```
DFS000I     REGID  JOBNAME   TYPE   TRAN/STEP  PROGRAM CLASS   IMSA
DFS000I            MSGRGN11  TP     WAITING                    IMSA
DFS000I            BATCHREG  BMP    NONE                       IMSA
DFS000I            FPRGN     FP     NONE                       IMSA
DFS000I            DBRCJBB1  DBRC                              IMSA
DFS000I            DLIJBB1   DLS                               IMSA
DFS000I     VTAM ACB OPEN           -LOGONS ENABLED            IMSA
DFS000I     *86275/223130*   IMSXRF     ACTIVE                 IMSA
```

5 From the active subsystem IMS Master Terminal, display the status of all subsystems related to IMS. The status of DB2 is:

```
DFS000I     SUBSYS    CRC   REGID  PROGRAM  LTERM  STATUS   IMSA
DFS000I     DSN       -                            CONN     IMSA
```

6 Start IMS from the alternate subsystem MVS console.

7 From the alternate subsystem IMS Master Terminal, issue:

```
/ERE backup
```

in order to define System 2 as the alternate IMS subsystem.

8 From the alternate subsystem IMS Master Terminal, start the VTAM ACB.

9 From the alternate subsystem IMS Master Terminal, start the IMS Message Processing Region.

10 From the alternate subsystem IMS Master Terminal, display what is active for the alternate IMS subsystem.

```
DFS000I     REGID  JOBNAME   TYPE   TRAN/STEP  PROGRAM CLASS   IMSB
DFS000I            MSGRGN21  TP     WAITING FOR SWITCHOVER     IMSB
DFS000I            BATCHREG  BMP    NONE                       IMSB
DFS000I            FPRGN     FP     NONE                       IMSB
DFS000I            DBRCJBB2  DBRC   WAITING FOR SWITCHOVER     IMSB
DFS000I            DLIJBB2   DLS                               IMSB
DFS000I     VTAM ACB OPEN           -LOGONS DISABLED           IMSB
DFS000I     *86275/223513*   IMSXRF    BACKUP - TRACKING       IMSB
```

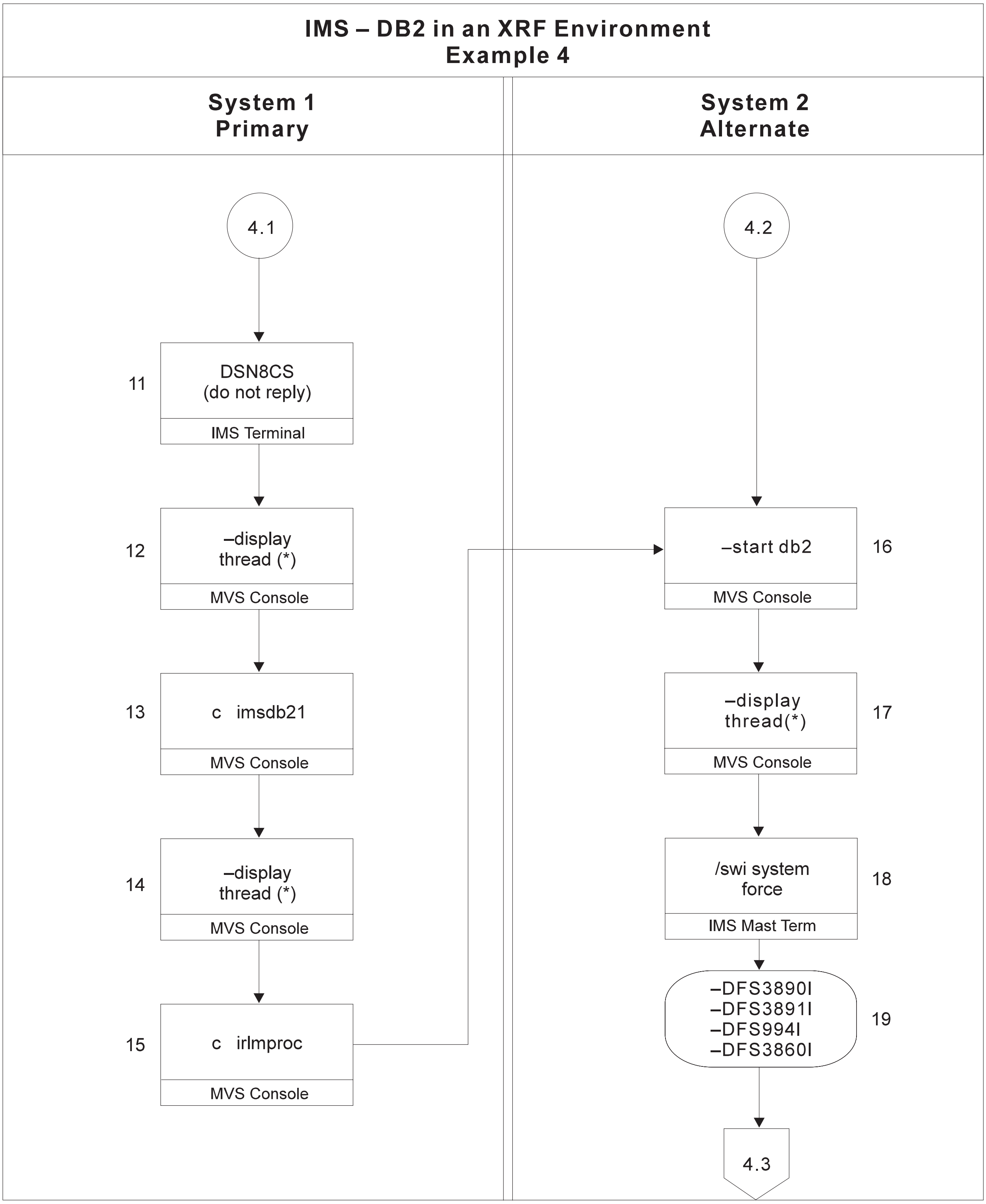

Figure 32. DB2/XRF Forced Takeover, Cancel IMS/IRLM (contd.)

Forced Takeover, Cancel IMS/IRLM (contd.)

11 From one Active subsystem IMS terminal, issue the sample transaction "DSN8CS." (Note that this transaction has been modified in order to keep it in-flight during the takeover tests.)

12 Issuing -DISPLAY THREAD(*) from the active subsystem MVS console gives the following:

```
DSNV401I - DISPLAY THREAD REPORT FOLLOWS -
DSNV402I - ACTIVE THREADS
NAME     ST A   REQ ID           AUTHID   PLAN       ASID
IMSXRF   T       56 0001DSN8IC02 MTO      DSN8IC02   0013
IMSXRF   N        2              IBMUSER             0027
DISPLAY ACTIVE REPORT COMPLETE
```

13 In order to force the takeover, cancel IMS. We then receive the DB2 related messages:

```
DFS3611I EXTERNAL SUBSYSTEM DSN     CONNECTION TERMINATED RC=B.  IMSA
DSN3201I - ABNORMAL EOT IN PROGRESS FOR USER=IBMUSER
           CONNECTION-ID=IMSXRF CORRELATION-ID=
DSN3201I - ABNORMAL EOT IN PROGRESS FOR USER=MTO
           CONNECTION-ID=IMSXRF CORRELATION-ID=0001DSN8IC02
*AVM006E TELL OPERATOR AT BACKUP TO REPLY "GO" TO MESSAGE AVM005A.
         I/O PREVENTION IS COMPLETE FOR SUBSYSTEM IMSA,
         FAILING ACTIVE ELEMENT OF RSE IMSXRF
```

This probably backed out the thread so that there was nothing to resolve when DB2 restarted.

14 -DISPLAY THREAD (*) command gives the following results:

```
DSNV401I - DISPLAY THREAD REPORT FOLLOWS -
DSNV419I - NO ACTIVE CONNECTIONS FOUND
```

15 Cancel the DB2 IRLM procedure.

16 From System 2 (alternate) MVS console, issue -START DB2.

During DB2 startup, the main following messages occurred:

```
DSNY001I - SUBSYSTEM STARTING
IEC161I 056-084,DSNMSTR,DSNMSTR,BSDS1,,,DSNC120.bsds01,
IEC161I DSNC120.BSDS01.DATA,CATALOG.IXRFDB2
        (these 2 lines are repeated for data and index of both
         BSDS copies. They are the well-known VSAM verify messages
         issued at OPEN time when a previous CLOSE has not been
         successful.)
DSNR001I - RESTART INITIATED
DSNR003I - RESTART...PRIOR CHECKPOINT RBA=000000F28060
DSNR004I - RESTART...UR STATUS COUNTS
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN ABORT=0
DSNR005I - RESTART...COUNTS AFTER FORWARD RECOVERY
IN COMMIT=0, INDOUBT=0
DSNR006I - RESTART...COUNTS AFTER BACKWARD RECOVERY
INFLIGHT=0, IN ABORT=0
DSNR002I - RESTART COMPLETED
DSN9022I - DSNYASCP -START DB2 NORMAL COMPLETION
```

17 A -DISPLAY THREAD(*) command issued from the alternate System MVS console gives:

```
DSNV419I - NO ACTIVE CONNECTIONS FOUND
```

Steps 18 and 19 refer to Figure 32 on page 358. Steps 20 through 24 refer to Figure 33 on page 361.

18 From the alternate subsystem IMS Master Terminal, issuing the `/SWI SYSTEM FORCE` command will force the takeover.

19 We will then get the following major messages:

```
*DFS3890I TAKEOVER REQUESTED REASON CODE = 78
*DFS3891I TAKEOVER IN PROGRESS
DFS994I TAKEOVER COMPLETED
DFS3860I 22:45:39 ALL TERMINAL SESSIONS SWITCHED. IMSB
```

20 We can now reply "GO" to the message:

```
AVM005A REPLY GO WHEN I/O PREVENTION COMPLETES FOR RSE IMSXRF
```

As a result, we will get the messages:

```
DFS0488I - UNLOCK COMMAND COMPLETED. RC = 00  IMSB
DFSAOUE0: DFS0488I - UNLOCK COMMAND COMP
```

21 **From now on, System 2 is the new active.**

22 Issue the `/DIS A` IMS command:

```
DFS000I     REGID  JOBNAME   TYPE   TRAN/STEP  PROGRAM CLASS   IMSB
DFS000I            MSGRGN21  TP     WAITING                    IMSB
DFS000I            BATCHREG  BMP    NONE                       IMSB
DFS000I            FPRGN     FP     NONE                       IMSB
DFS000I            DBRCJBB2  DBRC                              IMSB
DFS000I            DLIJBB2   DLS                               IMSB
DFS000I     VTAM ACB OPEN           -LOGONS ENABLED            IMSB
DFS000I     *86275/224813*   IMSXRF    ACTIVE                  IMSB
```

23 The "DSN8CS" transaction is backed out and re-executed automatically, giving a -924 SQL code:

```
DB2 CONNECTION INTERNAL ERROR,
```

which means that no connection exists between IMS and DB2. In order to get things going again, issue `/STA SUBSYS DSN`.

24 The first panel of "DSN8CS" transaction is displayed. This is part of the transaction's design.

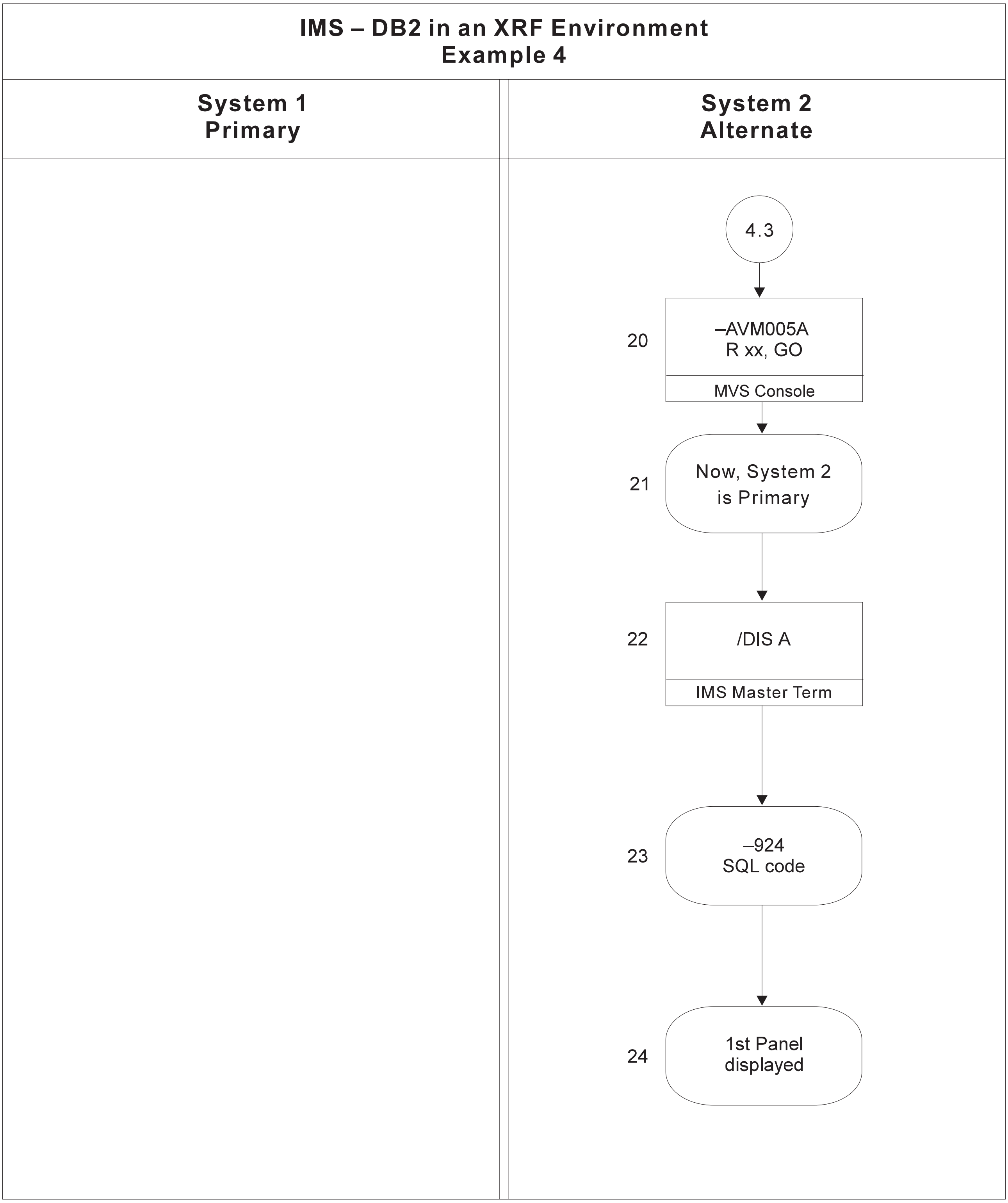

Figure 33. DB2/XRF Forced Takeover, Cancel IMS/IRLM (contd.)

Appendix D. Installation Worksheets

The purpose of these worksheets is to assist in planning and customizing IMS Automation. The format of the worksheets will assist you in asking questions, and provides space for answers to be used for reference during the actual IMS Automation installation.

Planning Worksheets

NetView Domains and IMS Regions

MVS system __________ NetView domain __________

IMS control region _______ Job name _______ Subsystem name _______

Use this table to record the names of all the dependent regions under this control region.

MVS job name	Subsystem name	Subsystem name	MVS job name	Subsystem name	Subsystem name
_____________	_____________	_____________	_____________	_____________	_____________
_____________	_____________	_____________	_____________	_____________	_____________
_____________	_____________	_____________	_____________	_____________	_____________
_____________	_____________	_____________	_____________	_____________	_____________
_____________	_____________	_____________	_____________	_____________	_____________

MVS system __________ NetView domain __________

IMS control region _______ Job name _______ Subsystem name _______

Use this table to record the names of all the dependent regions under this control region.

MVS job name	Subsystem name	Subsystem name	MVS job name	Subsystem name	Subsystem name
_____________	_____________	_____________	_____________	_____________	_____________
_____________	_____________	_____________	_____________	_____________	_____________
_____________	_____________	_____________	_____________	_____________	_____________
_____________	_____________	_____________	_____________	_____________	_____________
_____________	_____________	_____________	_____________	_____________	_____________

MVS system __________ NetView domain __________

IMS control region _______ Job name _______ Subsystem name _______

Use this table to record the names of all the dependent regions under this control region.

MVS job name	Subsystem name	Subsystem name	MVS job name	Subsystem name	Subsystem name
_____________	_____________	_____________	_____________	_____________	_____________
_____________	_____________	_____________	_____________	_____________	_____________
_____________	_____________	_____________	_____________	_____________	_____________
_____________	_____________	_____________	_____________	_____________	_____________
_____________	_____________	_____________	_____________	_____________	_____________

IMS Subsystem Information Sheets

For domain _______________

Subsystem name	VTAM node	Generic Application ID (XRF only)	Specific Application ID	Parent subsystem

Automation Operator Information Sheets

Use the following to define the automation operators used specifically for IMS automation.

Automation operators for domain _______________

AUTOOPS Keyword	Operator ID	IMS Subsystem	Purpose
IMSMSTR=		N/A	
IMSCPPI=		N/A	
IMSOP01=	____________	____________	
IMSOP02=	____________	____________	
IMSOP03=	____________	____________	
IMSOP04=	____________	____________	
IMSOP05=	____________	____________	
IMSOP06=	____________	____________	
IMSOP07=	____________	____________	
IMSOP08=	____________	____________	
IMSOP09=	____________	____________	
IMSOP10=	____________	____________	
IMSOP11=	____________	____________	
IMSOP12=	____________	____________	
IMSOP13=	____________	____________	
IMSOP14=	____________	____________	
IMSOP15=	____________	____________	
IMSOP16=	____________	____________	
IMSOP17=	____________	____________	

IMS Subsystem Group Information Sheet

IMS Automation lets you put IMS subsystems into groups. You then can execute commands to all the subsystems in a group with a single command. If you have defined a single-point-of-control, you can start or stop groups of subsystems across NetView domains.

Group name	IMS subsystems for this group

Service Period Information Sheets

In the form that follows, enter the default times at which you want each IMS subsystem to start and stop each day of the week. You can define up to five time frames for each day.

IMS subsystem name: ______________________

Day	Start/Stop	Start/Stop	Start/Stop	Start/Stop	Start/Stop
Daily					
Weekday					
Weekend					
Monday					
Tuesday					
Wednesday					
Thursday					
Friday					
Saturday					
Sunday					

IMS subsystem name: ______________________

Day	Start/Stop	Start/Stop	Start/Stop	Start/Stop	Start/Stop
Daily					
Weekday					
Weekend					
Monday					
Tuesday					
Wednesday					
Thursday					
Friday					
Saturday					
Sunday					

IMS subsystem name: ______________________

Day	Start/Stop	Start/Stop	Start/Stop	Start/Stop	Start/Stop
Daily					
Weekday					
Weekend					
Monday					
Tuesday					
Wednesday					
Thursday					
Friday					
Saturday					
Sunday					

Triggers Worksheet

Use the following table to specify triggers to be defined in the control file. Up to 28 event names can be used for each subsystem. Read "TRIGGER—Startup and shutdown triggers" on page 142 before filling out this information.

IMS subsystem name: ______________________

Check one	SERVICE (yes/no)	Event name	Event name	Event name	Event name	Event name	Event name
__STARTUP __SHUTDOWN							
__STARTUP __SHUTDOWN							
__STARTUP __SHUTDOWN							
__STARTUP __SHUTDOWN							
__STARTUP __SHUTDOWN							
__STARTUP __SHUTDOWN							
__STARTUP __SHUTDOWN							
__STARTUP __SHUTDOWN							
__STARTUP __SHUTDOWN							

IMS subsystem name: ______________________

Check one	SERVICE (yes/no)	Event name	Event name	Event name	Event name	Event name	Event name
__STARTUP __SHUTDOWN							
__STARTUP __SHUTDOWN							
__STARTUP __SHUTDOWN							
__STARTUP __SHUTDOWN							
__STARTUP __SHUTDOWN							
__STARTUP __SHUTDOWN							
__STARTUP __SHUTDOWN							
__STARTUP __SHUTDOWN							
__STARTUP __SHUTDOWN							

System IPLs, NetView Restarts, and IMS Restarts

Schedule the following:

Event	Description	Resource name	When
MVS IPL	Your MVS system will need to be IPLed after the IEAAPFxx member has been changed. Do this before NetView is restarted.		
NetView restart	This restart takes place after the customization steps are complete.		
IMS restart	If a new library was added as a result of the basic IMS definitions, the IMS system needs to be restarted. The IMS nucleus may need to be regenerated to include IMS Automation's A0I exit. A RESTART following the regen would be required.		

Event	Description	Resource name	When
MVS IPL	Your MVS system will need to be IPLed after the IEAAPFxx member has been changed. Do this before NetView is restarted.	________	________
NetView restart	This restart takes place after the customization steps are complete.	________ ________ ________ ________ ________ ________ ________	________ ________ ________ ________ ________ ________
IMS restart	If a new library was added as a result of the basic IMS definitions, the IMS system needs to be restarted. The IMS nucleus may need to be regenerated to include IMS Automation's A0I exit. A RESTART following the regen would be required.	________ ________ ________ ________ ________ ________ ________ ________ ________ ________ ________ ________ ________ ________ ________ ________ ________ ________ ________ ________	________ ________ ________ ________ ________ ________ ________ ________ ________ ________ ________ ________ ________ ________ ________ ________ ________ ________ ________ ________

Customization Worksheets

SA OS/390 Requirement Worksheet

STATUS FILE	
DOMAIN	______________________
DSN	______________________
PRTNRDOMAIN	______________________
PRTNRDSN	______________________
STATUS FILE	
DOMAIN	______________________
DSN	______________________
PRTNRDOMAIN	______________________
PRTNRDSN	______________________
STATUS FILE	
DOMAIN	______________________
DSN	______________________
PRTNRDOMAIN	______________________
PRTNRDSN	______________________
STATUS FILE	
DOMAIN	______________________
DSN	______________________
PRTNRDOMAIN	______________________
PRTNRDSN	______________________
STATUS FILE	
DOMAIN	______________________
DSN	______________________
PRTNRDOMAIN	______________________
PRTNRDSN	______________________
Note: The IMS STSFILE entry is to be coded exactly as noted in "STSFILE—Status file" on page 131.	

Glossary of IMS Automation Terms

This glossary defines special IMS terms used in the library and words used with other than their everyday meaning. In some cases, a definition may not be the only one applicable to a term, but it gives the particular sense in which it is used in the IMS Automation Option library.

abend. Abnormal end of task.

ACB. Access Method Control Block (VTAM and VSAM).

access method. A technique for moving data between main storage and input/output devices.

ANSI. American National Standards Institute.

AOST. Automated Operator Station Task.

APAR. Authorized program analysis report.

application program. A program written for or by a user that applies to the user's work. In data communication, a program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

automation. Computer system control of operation processes.

authorized program analysis report (APAR). A request for correction of a problem caused by a defect in a current unaltered release of a program.

batch. An accumulation of data to be processed.

batch message processing. In IMS/VS, a batch processing program that accesses online data bases and message queues.

BMP. Batch Message Processing Region.

CCTL. Coordinator Controller.

central processing complex (CPC). A conglomeration of several processors and other devices in one or more physical units. This usually means several processors running under the control of a single MVS/ESA operating system. For example, a 3090 model 400 processor complex can run as a four-processor CPC, or it can be partitioned into the equivalent of two 3090 model 200s, each of which runs as a CPC with its own operating system.

CICS. Customer Information Control System.

CLIST. Command List.

CMM. Critical Message Manager.

CNM. Communications Network Management.

command. In IMS, an instruction similar in format to a high-level programming language statement.

command list (CLIST). A list of commands and statements designed to perform a specific function for the user. Command lists can be written in REXX or in NetView Command List Language.

common state handler (CSH). Routine that IMS Automation calls from the NetView Automation Table to drive the actions defined in state/action tables.

concurrent. Pertaining to the occurrence of two or more activities within a given interval of time.

control file. In IMS Automation, the SA OS/390 file where a system programmer defines the automation environment, resources to be automated, and the extent and characteristics of automation.

control file entry. An entry in the control file that carries out specific functions on the basis of defined parameters and keywords associated with that entry. Some control file entries are unique to IMS Automation.

CPC. Central Processing Complex.

critical message manager (CMM). Facility in IMS Automation which displays critical messages in a scrollable format and enables operators to access information relating to the critical messages displayed.

CSA. Common Storage Area.

CSH. Common State Handler.

database. A collection of data fundamental to a system.

database backout. The function of removing changes made to user data sets by in-flight transactions.

database recovery. The function of restoring the user data sets, starting with a backup copy and applying all changes made to each data set after the backup was taken.

data security. The protection of data against unauthorized disclosure, transfer, modifications, or destruction, whether accidental or intentional.

data set. The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

DBCTL. Data Base Control.

DEDB. Data Entry Data Base.

DLISAS. Data Language Interface Separate Address Space (IMS Batch).

domain. In IMS, a set of subsystems on a specific NetView domain defined by the system programmer in the control file.

end user. In IMS, anyone using IMS to do a job, usually by interacting with an application program (transaction) by means of a terminal.

exception. An abnormal condition such as an I/O error encountered in processing a data set or a file, or using any resource.

Fast Path. IMS Automation function which enables the user to access any IMS Automation interface panel by entering = and an identifying number.

Fast Path Message Region. In IMS, a region that executes programs that require good response characteristics and that have large transaction volumes. Message processing is grouped for load balancing and synchronized for database integrity and recovery.

focal point system. In IMS, a system in which multiple subsystems are interconnected. One subsystem serves as a focal point of control, and the others are referred to as intermediate or distributed systems.

group. In IMS, a set of subsystems defined under one group name by the system programmer in the control file.

HM. Help Message.

HSBID. Hot Standby Identifier.

HSSP. High-Speed Sequential Processing.

initial program load (IPL). The initialization procedure that causes an operating system to commence operation.

initialization. Actions performed by IMS to construct the environment in the IMS region to enable IMS applications to be run. A process started by SA OS/390 and IMS Automation to construct the environment in which automation will occur.

installation. A particular computing system, in terms of the work it does and the people who manage it, operate it, apply it to problems, service it and use the work it produces. The task of making a program ready to do useful work. This task includes generating a program, initializing it, and applying PTFs to it.

Installation Verification Procedure (INSTALL/IVP). Procedure distributed with the system that tests the newly generated system to verify that the basic facilities are functioning correctly.

INSTALL/IVP. Install/Installation Verification Procedure.

Integrated Resource Lock Manager (IRLM). In IMS Automation, this facility is used as a lock manager, both as a single lock manager and in a data sharing environment.

intercommunication facilities. A generic term covering intersystem communication (ISC) and multiregion operation (MRO).

intersystem communication (ISC). Communication between separate systems by means of SNA networking facilities or by means of the application-to-application facilities of an SNA access method. ISC links IMS systems, and it may be used for user application-to-user application communication, or for transparently executing IMS functions on a remote IMS system.

IPL. Initial Program Load.

IRC. Interregion communication.

IRLM. Integrated Resource Lock Manager.

ISC. Intersystem Communication.

IVP. Installation Verification Procedure.

keyword. A symbol that identifies a parameter. A part of a command operand that consists of a specific character string.

local. In data communication, pertaining to devices that are attached to a CPC by cables, rather than data links.

local device. A device, such as a terminal, whose control unit is directly attached to a computer's data channel. No data link is used. Contrast with remote device.

lock manager. Feature of IMS Automation responsible for serializing the recovery process in areas where multiple subsystems can invoke recovery actions.

member. See partitioned data set.

MPP. Message Processing Program.

MSC. Multiple Systems Coupling.

MSDB. Main Storage Data Base.

MTO. Master Terminal Operator.

Multiple Systems Coupling (MSC). An IMS/VS feature that permits geographically dispersed IMS/VS systems to communicate with each other.

NCCF. Network Communications Control Facility.

network. An interconnected group of nodes. The assembly of equipment through which connections are made between data stations.

network configuration. In SNA, the group of links, nodes, machine features, devices, and programs that make up a data processing system, a network, or a communication system.

Network Communications Control Facility (NCCF). IBM licensed program consisting of a base for command processors that can monitor, control, and improve network operations.

non-XRF (non-XRF IMS). Represent IMS in a non-XRF configuration.

NPDA. Network Problem Determination Aid/Application.

OLDS. Online Log Data Set.

online. Pertaining to a user's ability to interact with a computer. Pertaining to a user's access to a computer via a terminal.

panel. In IMS Automation, the set of information displayed on a single screen of the user interface.

parameter. (ISO) A variable that is given a constant value for a specified application and that may denote the application.

partitioned data set (PDS). A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data. Synonymous with program library.

parent. In IMS Automation, system programmers can define a *parent* subsystem such that the subordinate subsystems (children) cannot be started until the *parent* is active or a *parent* cannot be shut down until all its children/subsystems are inactive.

PDS. Partitioned Data Set.

platform control file (platform control file entries). Specifies that this is an entry or function of SA OS/390.

PPI. Program-to-program interface.

preprocessor. Routine in IMS Automation that enables the programmer to define unique GLOBALV names to store the state value of certain processes.

processor (ISO). In a computer, a functional unit that interprets and executes instructions.

Program-to-program interface (PPI). A NetView component used by IMS Automation to enable users to send or receive data buffers from other programs. It also allows system and application programs to send alerts to the NetView hardware monitor.

PTF. Program Temporary Fix.

PUT. Program update tape.

RACF. Resource Access Control Facility.

RDS. Restart Data Set.

RECON. Recovery Control.

recovery routine. A routine entered when an error occurs during the performance of an associated operation. It isolates the error, assesses the extent of the error, and attempts to correct the error and resume operation.

remote. In data communication, pertaining to devices that are connected to a data processing system through a data link.

remote device. A device, such as a terminal, connected to a data processing system through a data link.

remote system. In IMS intercommunication, a system that the local IMS system accesses via intersystem communication or multiregion operation.

resource. Any facility of the computing system or operating system required by a job or task, including main storage, input/output devices, the processing unit, data sets, and control or processing programs.

Resource Access Control Facility (RACF). A licensed program that provides for access control by identifying and verifying users to the system, authorizing access to DASD data sets, logging detected unauthorized access attempts, and logging detected accesses to protected data sets.

RMF. Resource Management Facility.

roll. In IMS Automation, the option to begin/rollover to another NetView session. This action is assigned to the PF6 key.

SDF. Status Display Facility. The display facility for SA OS/390.

security. Prevention of access to or use of data or programs without authorization.

service. The carrying out of effective problem determination, diagnosis, and repair on a data processing system or software product.

service periods. In IMS Automation, the time between the initiation of startup and shutdown automation for specific IMS subsystems.

single-point-of-control. Feature of IMS Automation enabling the operator to monitor and control IMS subsystems from a single NetView console.

SLDS. System Log Data Set.

SMU. Security Maintenance Utility.

SNA. Systems Network Architecture.

software. (ISO) Programs, procedures, rules, and any associated documentation pertaining to the operation of a computer system. Contrast with hardware.

startup. The operation of starting up IMS by the system operator.

state/action table. In IMS Automation, state/action tables are a matrix of system events, states, and actions created by the system programmer. When an event occurs, the system references the state/action table and takes appropriate action.

status. In IMS Automation, the state of an application—active, inactive, starting, stopping, recovering, and so on—at a moment in time.

status code. In IMS/VS, a two-character code in the program communication block (PCB) mask that indicates the results of a DL/1 call.

status file. In IMS Automation, a file used to log startup and shutdown information.

subsystem. A secondary or subordinate system. A resource defined to SA OS/390 and IMS Automation.

system. In IMS, an assembly of hardware and software capable of providing the facilities of IMS for a particular installation.

system initialization table. A table containing user-specified data that will control a system initialization process.

systems network architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of networks.

task. (ISO) A basic unit of work to be accomplished by a computer. Under IMS, the execution of a transaction for a particular user.

TCO. Timer-Controlled Operations.

terminal. A point in a system or communication network at which data can either enter or leave. In IMS, a device, often equipped with a keyboard and some kind of display, capable of sending and receiving information over a communication channel.

terminal operator. The user of a terminal.

transaction. A transaction may be regarded as a unit of processing (consisting of one or more application programs) initiated by a single request, often from a terminal. A transaction may require the initiation of one or more tasks for its execution.

trigger. A set of conditions defined in the control file as required conditions for startup or shutdown initialization on a particular IMS Automation subsystem.

update. To modify a file with current information.

VSCR. Virtual Storage Constraint Relief.

VTAM. Virtual Telecommunications Access Method. VTAM is one of the ways IMS communicates with terminals.

WTOR. Write To Operator with Reply.

XRF. Extended recovery facility, a software function that minimizes the effects of various failures on the end users.

Index

Special Characters

$PI 150
$SI 151
$TI 152
%INCLUDE 149

A

abend codes
  ABCODEPROG 45
  ABCODES/ACTCODES/ALTCODES 47
  ABCODETRAN 49
ACORESTART 51
adding regions 203
ALTRST 79
ALTSTART 79
AOI exit 8
APPLID 88
APPLVER 78
AREA 52
assembling REXX environment table 213
automated operator profile 196
automation
  operators 4
automation operator
  enhancements
    ACORESTART 51
    AUTOOPS (SA OS/390) 53
    AUTOOPS (subsystem) 54
    ENVIRON 76
    ENVIRON SETUP 83
    EVIEX002 154
    EVIEX003 156
    EVIEX004 163
    IMSCNTL 88
    IMSRSENM 92
    INITSTART 93
    RECOVERY 104
    RESTART 107
    STARTUP 126
    STSFILE 131
    SUBSYSTEM 133
  platform
    control file 3, 5
    ENVIRON SETUP 83
    status file 6
  required control file entries 202
AUTOOPS (SA OS/390) 53
AUTOOPS (subsystem) 54

B

BROADCAST (BRO) 55
BTIMER 78
build message processor 165

C

CHE 56
checking security 38, 179
CMM
  coding SDF 30
  IMS Automation interface option 12
  keyword 110
  overview 10
CODE—IMS SDF support 90
conditions, trigger 84
control facility
  EVIDMN 274
  EVIS2CFG 276
  EVIS2CQS 323
  EVIS2CRT 325
  EVIS2DBC 288
  EVIS2FDR 322
  EVIS2IMS 279
  EVIS2LCL 326
  EVIS2OPS 278
  EVIS2SPX 312
  EVIS2STA 277
  EVIS2XRF 295
  EVISTSM 275
control file 5
  definitions 201
  entries
    $PI 150
    $SI 151
    $TI 152
    ABCODEPROG 45
    ABCODES 47
    ABCODETRAN 49
    abend codes 47
    ACORESTART 51
    ACTCODES 47
    ALTCODES 47
    AREA 52
    AUTOOPS 53
    BROADCAST (BRO) 55
    CHE 56
    CODE keyword on IMS CRITMSGS 90
    CQS0031A 57
    CQS0032A 58
    CQS0033A 59
    CQSET 60

control file *(continued)*
entries *(continued)*
DFS3869A/DFS3869I 66
DFS810A 69
DFS994I 73
ENVIRON 76
ENVIRON SETUP 83
EVIEX002 154
EVIEX003 156
EVIEX004 163
EXTCOND 84
FORCE 86
HOLDQ 87
IMSCNTL 88
IMSGROUP 91
IMSRSENM 92
INITSTART 93
IPS 95
NTFYOP 96
OLDS 97
POSTCHKP 100
PRECHKP 101
PRODUCT 102
RECONS 103
RECOVERY 104
RELEASEQ 106
RESTART 107
RESTARTABORT 108
SDF 110
SDF support 110
SERVICE 112
SHUTFORCE 115, 117
SHUTIMMED 115, 117
SHUTINIT 114
SHUTNORM 115, 117
SNAPQ 123
START1 124
START2 124
STARTUP 126
STOPBMPREGION 128
STOPFPREGION 129
STOPREGION 130
STSFILE 131
SUBSYSTEM 133
TCO 136
TCOMEMBERS 137
TPABEND 141
TRIGGER 142
TRIGGERS 142
UNLKAVM 144
UNLOCK 145
UP 146
entry format 41
control file definitions, installing 201
control of IMS 9
Control Region automation statements
$PI 150
$SI 151
$TI 152
ABCODEPROG 45
ABCODETRAN 49
BROADCAST (BRO) 55
CHE 56
CQS0031A 57
CQS0032A 58
CQS0033A 59
CQSET 60
DFS3869A/DFS3869I 47
FORCE 86
HOLDQ 87
IPS 95
POSTCHKP 100
PRECHKP 101
RELEASEQ 106
SHUTFORCE 115, 117
SHUTIMMED 115, 117
SHUTINIT 114
SHUTNORM 115, 117
SNAPQ 123
START1 124
START2 124
STOPBMPREGION 128
STOPFPREGION 129
STOPREGION 130
TCO 136
TCOMEMBERS 137
TPABEND 141
UNLKAVM 144
UNLOCK 145
UP 146
VTAMTERMS 148
CQS0031A 57
CQS0032A 58
CQS0033A 59
CQSET 60
Critical Messages Manager
coding SDF 30
IMS Automation interface option 12
keyword 110
overview 10
customization
IMS Automation 201
SDF 214
transaction recovery 34

D

DB2
forced takeover sequence 346, 355
general scenario 332
normal takeover sequence 338

DB2 *(continued)*
- recommendations 331

DEFFORCE 77
defining
- notification operator 96
- startups and shutdowns 30
- transaction recovery 34

DEFLTHSB 89
DEFSTART 77
DEFXRFST 78
DFS2142 61
DFS2161I 63
DFS2169I 64
DFS3258A 65
DFS3869A/DFS3869I 66
DFS5541A 67
DFS810A 69
DFS989I 71
DFS994I 73
DFSMDA dynamic allocation macro 98
dialog, when to use
- IMS Automation definitions 201

display facility
- CODE 90
- coding, basic overview 29
- critical messages 30, 90
- customizing 214
- installing 194
- messages panel 29
- panels 27
- support 110
- using 25

DSIDMN 193, 196
DSIOPF 196
DSIPARM 193, 194
DSIPRF 196

E

ENVIRON 76
ENVIRON SETUP 83
environment table 213
EVICMD 270
EVIEX002 154
EVIEX003 156
EVIEX004 163
EVIMAVM1 250
EVIMCON0 246
EVIMCON1 251
EVIMCQS1 264
EVIMDFS1 253
EVIMDSP1 262
EVIMDXR1 263
EVIMEVI0 247
EVIMEVI1 266
EVIMIOS1 268
EVIMSG00 245
EVIMSG01 248
EVINTASK
- sample member 243

EVINTASK - NetView PPI initialization member 222
EVIOPF 269
EVIPRFAO 196, 329
EVIS2CFG 276
EVIS2CQS 323
EVIS2CRT 325
EVIS2DBC 288
EVIS2FDR 322
EVIS2IMS 279
EVIS2LCL 326
EVIS2OPS 278
EVIS2SPX 312
EVIS2STA 277
EVIS2XRF 295
EVISDMN 274
EVISECUR 273
EVISJ010 sample JCL 232
EVISJ020 sample JCL 234
EVISJ021 213
EVISPINM 217, 239
EVISPINM PPI member 217
EVISROUT—IMS message router 181
EVISTSM 275
examples
- IPDS Initialization Table entries 239
- NetView automation table entries 245, 246, 247, 248, 250, 251, 253, 262, 263, 264, 266, 268
- NetView operator definition 269
- NetView PPI initialization member 243
- required Command Model entries 270, 273
- sample configuration tables 274, 276, 277, 278, 279, 288, 295, 312, 322, 323, 325, 326
- sample DSTINIT 275
- sample JCL 231

exit, AOI 8
EXITS parameter
- ENVIRON SETUP 83

EXTCOND 84
extended control file definitions 210
external conditions 84

F

focal point 24
FORCE 86
forwarding a message 170
FTIMER 78

G

grouping IMS subsystems
  IMSGROUP control file entry 91
  overview 30

H

hierarchy 30
HOLDQ 87

I

IMS
  configuration information 274, 276, 277, 278, 279, 288, 295, 312, 322, 323, 325, 326
  initialization table 217
  message router 181
IMSAPPL 88
IMSARCH keyword 110
IMSBMSG 165
IMSCMD 167
IMSCNTL 88
IMSFWM 170
IMSGROUP 91
IMSMSCL keyword 110
IMSMSG00 245
IMSMSG01 248
IMSOLDS keyword 110
IMSPOST 172
IMSQRY 174
IMSRCMD 178
IMSRECN keyword 110
IMSRSENM 92
IMSSEC 179
IMSSTRCT keyword 110
IMSTIMR keyword 110
IMSTRAN keyword 110
including additional members 149
initialization member, program-to-program interface
  EVISPINM 217
  IMS 199, 217
INITSTART 93
Installation Verification Procedure (IVP) 225
installing
  control file definitions 201
  environment table 213
  EVISPINM 217
  extended control file definitions 210
  IMS Automation 189
  initialization member 199, 217
  maintain IMS messages 217
  program-to-program interface 217
  REXX environment table 213
  SDF 214
  security checking 209
invoking security checking 179
IPS 95
issuing IMS commands 167
ITIMER 78
IVP (Installation Verification Procedure) 225

J

JCL samples 231, 235

L

local application support 182

M

maintain IMS messages 217
MAJNODE1 88
MAJNODE2 88
merging the control file 194
message automation statements
  DFS2142 61
  DFS2161I 63
  DFS2169I 64
  DFS3258A 65
  DFS3869A/DFS3869I 66
  DFS554I 67
  DFS810A 69
  DFS989I 71
  DFS994I 73
  RESTARTABORT 108
message region test scenario 227
message router 181
monitor IMS 9

N

name lookup 174
NetView
  automation table
    EVIMAVM1 250
    EVIMCON0 246
    EVIMCON1 251
    EVIMCQS1 264
    EVIMDFS1 253
    EVIMDSP1 262
    EVIMDXR1 263
    EVIMEVI0 247
    EVIMEVI1 266
    EVIMIOS1 268
    EVIMSG00 245
    EVIMSG01 248
    overview 4
    required entries 239, 245, 246, 247, 248, 250, 251, 253, 262, 263, 264, 266, 268
  domain worksheet 364
  profile data set 196

NetView PPI initialization member 222
notation conventions, control file entries 44
NTFYOP 96

O

OLDS control file entry 97
OLDS, control file entries for 207
operator
  definition member 196
  interface 8
  notification 96
OTHMAJ1 89
OTHMAJ2 89

P

panel support 182
parameters
  data sets 193
PARMS1 89
PARMS2 89
planning
  installation 187
POSTCHKP 100
posting an external event 172
PPI initialization member 222
PRECHKP 101
PRODUCT 102
profile
  automated operators 196
  data set, NetView 196
  data sets 193
program-to-program interface initialization member 222
  avoiding problems with 191
  EVISPINM 217
  IMS 199, 217
PRTNRDOM 79
PRTNRID 79
PRTNRSUB 79

R

RECONS control file entry 103
RECONs, control entries for 208
recovery 104
  overview 16
  transaction, defining 34
RELEASEQ 106
request an IMS function 178
RESTART 107
RESTARTABORT 108
REXX environment table 213
RTIMER 78

S

SDELAY 78
SDF
  CODE 90
  coding, basic overview 29
  critical messages 30, 90
  customizing 214
  installing 194
  messages panel 29
  panels 27
  support 110
  using 25
SDF support 110
security checking 209
  classifying operators 38
  IMSSEC 179
  installing 209
  invoking 179
SERVICE 112
service periods
  control file entry 112
  defining 31
  worksheets 368
SHUTDLY 133
shutdown, subsystem
  defining 30
  overview 15
SHUTFORCE 115, 117
SHUTIMMED 115, 117
SHUTINIT 114
SHUTNORM 115, 117
SHUTOPTIONS 134
SHUTTYPES 120
single-point-of-control 8, 24
SNAPQ 123
START1 124
START2 124
STARTCMD 133
STARTOPTIONS 133
STARTUP 126
startup, subsystem
  defining 30
  overview 14
  triggers 142
state/action tables
  AREA 52
  control file entry 52, 102
  overview 7
  PRODUCT 102
  using 36
status file 6
STOPBMPREGION 128
STOPFPREGION 129
STOPREGION 130

STSFILE 131
SUBID 77
SUBSYSTEM 133
subsystems
  group worksheets 367
SUBTYPE 76
SYNCH 77
syntax rules 44
system hierarchy 30

T

takeover, XRF
  overview 18
  reason code 11
  VTAMTERMS 148
TCO 136
TCOMEMBERS 137
testing 225, 227
THRESHOLDS, IMS error 139
thresholds, transaction recovery 35
TKO 79
TPABEND 141
transaction
  abend recovery 49
  recovery 34
transaction recovery, control file entries 208
TRIGGER 142
triggers
  conditions 84
  control file entry 142
  defining 142
  posting events 172
  setting conditions 172
TTIMER 78

U

UNLKAVM 144
UNLOCK 145
UP 146
USERSTART 147

V

VTAMTERMS 148

W

WDELAY 78
worksheets
  automation operator 366
  NetView domains and IMS regions 364
  SA OS/390 372
  scheduling restarts 370
  service periods 368
  subsystem groups 367

worksheets *(continued)*
  subsystems 365
WTCHALERT 80
WTCHNOTF 79
WTIMER 78

X

XRF subsystems
  ENVIRON entry 77
  environment 13
  takeover conditions 18
  VTAMTERMS 148

Communicating Your Comments to IBM

System Automation for OS/390
AOC/MVS IMS Automation
Programmer’s Reference
and Installation Guide
Version 1 Release 4

Publication No. SC23-3817-02

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of the book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF form and either send it postage-paid in the United States, or directly to:

  IBM Deutschland Entwicklung GmbH
  Department 3248
  Schoenaicher Strasse 220
  D-71032 Boeblingen
  Federal Republic of Germany

- If you prefer to send comments by FAX, use this number:
  - (Germany): 07031-16-3456
  - (Other countries): (+49)+7031-16-3456
- If you prefer to send comments electronically, use this network ID:

  IBM Mail Exchange: DEIBMBM9 at IBMMAIL

  Internet: s390id@de.ibm.com

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies.

Readers' Comments — We'd Like to Hear from You

System Automation for OS/390
AOC/MVS IMS Automation
Programmer's Reference
and Installation Guide
Version 1 Release 4

Publication No. SC23-3817-02

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	□	□	□	□	□

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	□	□	□	□	□
Complete	□	□	□	□	□
Easy to find	□	□	□	□	□
Easy to understand	□	□	□	□	□
Well organized	□	□	□	□	□
Applicable to your tasks	□	□	□	□	□

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? □ Yes □ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

Phone No.

Readers' Comments — We'd Like to Hear from You
SC23-3817-02

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Attn: Dept EHJ - BP/003D
6300 Diagonal Highway
Boulder, CO 80301-9151

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

Program Number: 5685-151

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC23-3817-02