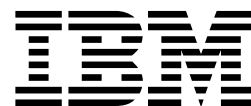




System Automation for OS/390

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

Version 1 Release 4



System Automation for OS/390

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

Version 1 Release 4

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 CICS Automation (5685-151), and to all subsequent releases and modifications until otherwise indicated in new editions.

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.

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

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1990, 1999. All rights reserved.**

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

Contents

Notices	xiii
Programming Interface Information	xiii
Trademarks	xiii
 About This Book	xv
Who Should Use This Book	xv
What's in This Book	xv
What's New in This Book?	xv
Related Publications	xvi
The SA OS/390 Library	xvi
Related Product Information for the Base Program	xvii
Related Product Information for Workstation Operations	xviii
 Part 1. Understanding CICS Automation	1
 Chapter 1. The Components of Automation	3
The NetView Automation Table	4
Automation Operators	4
The SA OS/390 Control File	5
The SA OS/390 Status File	5
The State/Action Tables	6
CICS Automation's Use of the Program-to-Program Interface	6
NetView Components	6
CICS Components	7
Communication Components	7
The Operator Interface	7
Sysplex Considerations	10
 Chapter 2. How to Implement and Use CICS Automation	11
How to Use Single-Point-of-Control	12
Added Capabilities	13
This Is How It Works	13
This Is Required	14
How to Set Up the Status Display Facility for CICS Automation	14
The CICS Automation Status Display Facility Panels	16
How to Use CICS Automation for Startups and Shutdowns	18
Parent/Child Relationships in a Startup or Shutdown	18
Defining the Service Periods	19
Defining Startup and Shutdown Triggers	19
Using Triggers in a Distributed Environment	21
Grouping CICS	22
How to Define Transaction Recovery	23
How to Use the State/Action Tables	24
How It Works	24
State/Action Table Control File Entries	27
How to Use Health Checking	28
Setting Up Health Check Programs	28
Using the Health Check Operator Interface	29
How to Use Link Monitoring	30

Defining the Links to CICS Automation	30
What Is Echoplexing	30
How Security Checking Works	31
Security Checking Using NetView	31
Security Checking Using CICS	32
Adding Local Applications to the CICS Automation Operator Interface	33
Using Linemode Functions	35
Health Checking	35
CICS Startup and Shutdown	35
SIT Override	35
Link Monitoring	36
Message options	36
Trace Options	36
CICSPOST	36
CEMTPI	36
How to Implement Remote Site Recovery for VSAM RLS (CICS TS Function Only)	37

Part 2. Programming Information 39

Chapter 3. Programming the Control File	41
Differences in AOC/MVS Release 1 and SA OS/390 1.3 Syntax	41
Upgrading from CICS Automation 1.3 to CICS Automation 1.4	41
Special Considerations when Coding Control File Entries	41
Restrictions	41
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
Base SA OS/390 Control File Entries	45
%INCLUDE—Include Additional Members in Control File	46
ACORESTART—NetView Restart Commands	47
Comments and Usage Notes	47
Examples of Usage	47
Automation Flags	48
Comments and Usage Notes	48
Examples of Usage	48
AUTOOPS—Automation Operators	50
Comments and Usage Notes	50
Examples of Usage	51
SDF—Status Display Facility Color Definitions	52
Comments and Usage Notes	52
Examples of Usage	52
ENVIRON SETUP—CICS Initialization Support	53
Examples of Usage	53
ENVIRON TIMEOUT—CICS CEMT Support	54
Comments and Usage Notes	54
Examples of Usage	54
Shutdown Control File Entries	55
Comments and Usage Notes	55
Examples of Usage	55
STARTUP—Startup Commands	56

Comments and Usage Notes	56
Examples of Usage	56
SUBSYSTEM—Subsystem Definition	57
Comments and Usage Notes	57
Examples of Usage	57
THRESHOLDS—Error Thresholds	58
Comments and Usage Notes	58
Examples of Usage	58
CICS Automation Control File Entries	59
ABCODESYSTM - System abend recovery	60
ABCODETRAN - Transaction abend recovery	62
AREA - Define a set of State/Action Tables	64
CICSCNTL - CICS extensions for SUBSYSTEM entry	66
CICSGROUP - Group of CICS subsystems	69
CONNECTION - Link monitoring definitions	70
EXTCOND - External conditions	73
HEALTHCHK - Health checking	75
LISTSHUT - Transaction purging during shutdown	76
PRODUCT - Subsystem State/Action Table sets	77
RCVRAUTO - Autoinstall time limit	79
RCVRSOS - Short-on-storage handler	80
RCVRTRAN - Transaction recovery	81
RCVRVIOL - Storage violation handler	82
SERVICE - Service period definitions	83
TRIGGER - Startup and shutdown triggers	85
 Chapter 4. CICS Automation Routines, Commands, and Definition	
Members	87
Subroutines	88
CICSBMSG - Build message processor	89
CICSQRY - Name lookup	91
CICSRCMD - Request a CICS function	95
CICSSEC - Invoke security checking	96
EVESTS - Status file interface	99
Commands and Common Routines	107
CEMTPPI - CEMT PPI short syntax	109
CICSDLY - Change the shutdown delay time	110
CICSPOST - Post an external event	111
CICSPURG - Purge transactions	112
CICRSYC - CICS resync	113
CICSSHUT - Shutdown processor	114
EVEED003 - Critical message handler for the Status Display Facility	116
EVEERDMP - CICS dump	117
EVESROUT - CICS message router	118
EVEEMIGR - Migrate subsystem to CICS/TS	119
EVEEY00S - Common state handler for State/Action Tables	120
CICSHLTH - Linemode Health Checking	122
CICSSTRT - Linemode CICS Startup	124
CICSSTOP - Linemode CICS Shutdown	126
CICSOVRD - Linemode SIT Override	128
CICSLM - Linemode Link Monitor	130
CICSMHDR - Linemode Message Options	133
CICSTRCE - Linemode Trace Options	135
CMASSHUT - CICSplex SM Address Space (CMAS) Shutdown	137

Definition Members	138
EVESPINM - CICS PPI initialization member	139
EVENTASK - NetView PPI initialization member	141
EVESCMT2 - Message exit table for CICS Version 2	143
EVESCMT3 - Message exit table for CICS Version 3 and higher	145

Part 3. Planning, Installation, and Customization 149

Chapter 5. Planning	151
Pre-Installation and Planning Steps	151
Designing Your Automation Environment	152

Chapter 6. Installation	153
Step 3: Updating IEAAPFxx in SYS1.PARMLIB	153
Step 4: Defining Subsystem Allocatable Consoles	154
Step 5: Check the Subsystem Name Table	154

Chapter 7. Merge NetView Related Members	155
Step 1: Add CICS Automation Data Sets to NetView JCL	155
Step 2: Copy CICS Automation Sample Members to the Target Library	156
Step 3: Merge Status Display Facility Members	156
Step 4: Include EVECFG into the Control File	156
Step 5: Include EVECMD into DSICMD	156
Step 6: Include and Update the NetView Automation Table	157
Step 7: Tailoring Your Network NetView for CDRM Recovery	157
Step 8: Include EVEOPF into DSIOPF	158
Step 9: Merge the NetView Profile Data Set	158
Step 10: Include EVEDMN into DSIDMN and Update	158

Chapter 8. Define CICS Automation to CICS	159
Step 1: SIT or Startup Overrides	159
Step 2: Program List Table Definitions	160
Step 3: Transaction List Table Definitions	160
Step 4: Define Consoles	161
Step 5: Transaction and Program Definitions	161
Method 1: Batch	162
Method 2: Online	163
Step 6: DFHRPL and the CICS Automation Library	182

Chapter 9. CICS Automation Definitions in NetView	183
Step 1: Basic CICS Automation Common Control File Definitions	183
Step 2: Basic CICS Subsystem Control File Definitions	184
Step 3: CICS Automation Security Checking	184
Step 4: Extended CICS Automation Common Control File Definitions	186
Step 5: Extended CICS Subsystem Common Control File Definitions	186
Step 6: The program-to-program interface Initialization Member (EVENTASK)	187
Step 7: The REXX Environment Table	187
Step 8: Customizing the Status Display Facility	187
Step 9: Adding a New CICS Automation Status File Record	190
Step 10: Installing CICSplex SM REXX API	190

Chapter 10. Extended CICS Definitions	191
Step 1: Health Check Program and Transaction Definitions	191

Step 2: Program-to-program interface Initialization Member (CICS)	191
Step 3: Add or Change the CICS Transient Data Messages	192
Step 4: Echoplex Back End Programs	193
Step 5: Security Considerations	193

Chapter 11. CICS Automation Definitions for CICSplex System Manager (CPSM)	195
Automating Coordinating Address Space (CAS) Startup and Shutdown	195
Automating CICSplex SM Address Space (CMAS) Startup and Shutdown	195

Part 4. Appendixes	197
-------------------------------------	-----

Appendix A. CICS Automation and the Program-to-Program Interface	199
Program-to-Program Interface Components in NetView and CICS	199
NetView Requests Using the Program-to-Program Interface	200
CONVERSE from NetView	200
SEND from NetView	202
CANCEL from NetView	202
CICS Requests Using the Program-to-Program Interface	203
CONVERSE from CICS	203
SEND from CICS	204
Programming Interface	205
EVESNCCI - NetView to CICS Communication Interface	207
EVESNRSP - Common Response Handler from CICS	212
EVESSCCI - CICS to NetView Communication Interface	213
EVEMPINT - EVESSCCI Parameter List Copy Book	216

Appendix B. Sample JCL	221
EVESJ010—Copy the Sample Members to DSIPARM	221
EVESJ015—CICS/MVS Definitions	223
EVESJ020—Assemble and Link Edit Definition Members	224
EVESJ022—Assemble and Link Edit Health Check Programs	226

Appendix C. Sample and Source	229
EVECFG—%INCLUDE Statement for CICS Automation Control File Entries	229
EVECFG01—Sample CICS Automation Control File Entries	229
CICSMSG0 (EVEMSG00)—Early NetView Initialization Automation Table	244
EVEMEVE0—EVE Messages (Included by CICSMSG0)	245
EVEMCON0—Messages in Conflict with AOC/MVS (Included by CICSMSG0)	246
CICSMSG1 (EVEMSG01)—Production NetView Initialization Automation Table	247
EVEMEVE1—EVE Messages (Included by CICSMSG1)	248
EVEMDFH1—DFH Messages (Included by CICSMSG1)	250
EVEMCON1—Messages in Conflict with AOC/MVS (Included by CICSMSG1)	264
EVEESCMT2—Message Exit Table for CICS Version 2	267
EVEESCMT3—Message Exit Table for CICS Version 3 and Higher	271
EVECMD—NetView Command Model Statements Member CICS Automation Entries	277
EVESECUR—Security Definitions (Included by EVECMD)	279
EVEDMN—NetView Domain Member CICS Automation Entries	280
EVEOPF—NetView Operator Definitions CICS Automation Entries	281
EVETREE—The Status Display Facility Tree Structure	282

EVEPNLS—The Status Display Facility Panels	283
EVEESA01—Short-on-Storage State/Action Table	284
EVEESA02—Storage Violation State/Action Table	286
EVEESA03—Autoinstall Problem Recovery State/Action Table	287
EVEESA04—VTAM ACB Problem Recovery State/Action Table	288
EVEPRFAO—CICS Automation Operator Profiles	291
EVECHLTH—Health Check Sample Program (COBOL Version)	292
EVEAREA—Layout of the Health Checking DFHCOMMAREA	295
EVENTASK—PPI Initialization Member (NetView)	296
EVEPINM—PPI Initialization Member (CICS)	298
EVEYU1—CPSM Messages (Included by EVEMSG01)	302
 Appendix D. CICS Automation SMF Record Layout	 305
 Appendix E. Worksheets	 307
NetView Domains and CICS Regions	307
Basic CICS Subsystem Information Sheets	309
Automation Operator Information Sheets	312
CICS Subsystem Group Information Sheet	315
Service Periods Information Sheets	317
Link Monitoring Information Sheets	319
Startup and Shutdown Trigger Information Sheets	322
Health Checking Information Sheets	325
System IPLs, NetView Restarts, and CICS Restarts	327
 Appendix F. Extra Check Lists	 331
Planning Check Lists	331
Installation Check Lists	332
Merging NetView Related Members Check Lists	333
CICS Definitions in NetView Check Lists	334
CICS Automation Definitions in NetView Check Lists	335
Extended CICS Definitions Check List	336
 Glossary of CICS and Other Terms	 337
 Index	 341

Figures

1.	The CICS Automation Main Menu	7
2.	Focal Point Definitions in a Distributed Environment	13
3.	Tree Structures of a Distributed Environment	14
4.	The CICS Monitor Panel	17
5.	The CICSPOST Command Syntax	20
6.	Short on Storage State/Action Table Example	25
7.	State/Action Table Keywords	26
8.	Health Checking panel	29
9.	Master Control File Example	43
10.	The EVESTS Command	99
11.	Program-to-program interface Components in NetView and CICS	199
12.	An EVESNCCI CONVERSE Request	201
13.	An EVESNCCI SEND Request	202
14.	An EVESCCCI CONVERSE Request	204
15.	An EVESCCCI SEND Request	205

Tables

1.	SA OS/390 Library	xvi
2.	Related Product Books	xvii
3.	Related Product Books	xviii
4.	Samples and Source Used for Customization	11
5.	Control File Configuration Data	43
6.	CICSMSG Return Codes	90
7.	CICSQRY Return Codes	92
8.	CICSQRY Input and Output Mapping	94
9.	CICSRCMD Return Codes	95
10.	CICSSEC Return Codes	98
11.	CICSPOST Return Codes	111
12.	EVEEMIGR Return Codes	119
13.	Planning Check List	151
14.	Installation Check List	153
15.	Merging the NetView Related Members Check List	155
16.	CICS Definitions Check List	159
17.	CICS Automation Definitions in NetView Check List	183
18.	Extended CICS Definitions Check List	191
19.	EVESSCCI CONVERSE Request Parameter List	213
20.	EVESSCCI Fields Returned to Caller from CONVERSE Request	214
21.	EVESSCCI SEND Request Parameter List	215

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 the

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:

AS/400	IBM	OS/390
AT	IMS	RACF
CICS	MVS/ESA	RMF
CICS/ESA	MVS/XA	SP
CICS/MVS	NetView	VTAM
DB2		

About This Book

Use this book to install, customize, and maintain the *IBM Automated Operations Control/MVS (AOC/MVS) CICS Automation Feature, Version 1 Release 4*. Hereafter, this book refers to AOC/MVS CICS Automation as SA OS/390 CICS Automation or simply CICS 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).

CICS Automation is a feature of System Automation for OS/390 that provides a simple and consistent way to monitor and control all of the CICS regions, both local and remote, within your organization. This automation feature automates, simplifies, and standardizes console operations and the management of component, application, and production related tasks.

Who Should Use This Book

This book was written for system programmers, system designers, and application designers who will implement CICS automation using CICS Automation.

Do not attempt to install or customize this CICS Automation if you do not have a programmer's understanding of NetView, CICS, SA OS/390, and CICS Automation, because most of the definitions take place in these programs. Also, you will modify JCL, command lists, and programs for some of the automation functions.

What's in This Book

This book contains the following:

Part 1, Understanding CICS Automation

Describes the basic components of CICS Automation, such as the operator interface and control tables.

Part 2, Programming Information

Describes the control file entries and the programming interfaces.

Part 3, Planning, Installation, and Customization

Explains how to install and customize CICS Automation.

The Appendixes

Show samples of CICS Automation install JCL, control file entries, NetView Automation Table entries, health check routines, State/Action Tables, and so on.

What's New in This Book?

Changes to this book since Release 3 document:

- Introduction of SA OS/390 as a supported environment. CICS Automation is now a feature of SA OS/390.
- CICS Automation support of Release 4 of AOC/MVS 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.
- Technical changes reflecting service updates.

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

Related Publications

The SA OS/390 Library

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

<i>Table 1. SA OS/390 Library</i>	
Title	Order Number
<i>SA OS/390 General Information</i>	GC28-1541
<i>SA OS/390 Licensed Program Specifications</i>	GC28-1540
<i>SA OS/390 Planning and Installation</i>	GC28-1549
<i>SA OS/390 Customization</i>	GC28-1566
<i>SA OS/390 Operations</i>	GC28-1550
<i>SA OS/390 Messages and Codes</i>	GC28-1569
<i>SA OS/390 Technical Reference</i>	GC28-1593
<i>AOC/MVS CICS Automation General Information</i>	GC23-3813
<i>AOC/MVS CICS Automation Operator's Guide</i>	SC23-3815
<i>AOC/MVS CICS Automation Programmer's Reference and Installation Guide</i>	SC23-3814
<i>AOC/MVS IMS Automation General Information</i>	GC23-3816
<i>AOC/MVS IMS Automation Operator's Guide</i>	SC23-3818
<i>AOC/MVS IMS Automation Programmer's Reference and Installation Guide</i>	SC23-3817
<i>AOC/MVS OPC Automation General Information</i>	GC23-3819
<i>AOC/MVS OPC Automation Operator's Guide and Scheduler's Reference</i>	SC23-3821
<i>AOC/MVS OPC Automation Programmer's Reference and Installation Guide</i>	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.

<i>Table 2 (Page 1 of 2). Related Product Books</i>	
Title	Order Number
<i>MVS/ESA MVS Configuration Program Guide and Reference</i>	GC28-1817
<i>MVS/ESA Planning: Dynamic I/O Configuration</i>	GC28-1674
<i>MVS/ESA Support for the Enterprise Systems Connection</i>	GC28-1140
<i>MVS/ESA Planning: APPC Management</i>	GC28-1110
<i>MVS/ESA Application Development Macro Reference</i>	GC28-1822
<i>MVS/ESA SP V5 System Commands</i>	GC28-1442
<i>MVS/ESA SPL Application Development Macro Reference</i>	GC28-1857
<i>NetView for MVS V3R1 Administration and Security Reference</i>	SC31-8045
<i>NetView for MVS V3R1 Automation Implementation</i>	SC31-8050
<i>NetView for MVS V3R1 Automation Planning</i>	SC31-8051
<i>NetView for MVS V3R1 Command Reference</i>	SC31-8047
<i>NetView for MVS V3R1 Customization Guide</i>	SC31-8052
<i>NetView for MVS V3R1 Customization: Writing Command Lists</i>	SC31-8055
<i>NetView for MVS V3R1 Installation and Administration Guide</i>	SC31-8043
<i>NetView for MVS V3R1 RODM and GMFHS Programming Guide</i>	SC31-8049
<i>NetView for MVS V3R1 User's Guide</i>	SC31-8056
<i>NetView for MVS V3R1 Tuning Guide</i>	SC31-8048
<i>OS/390 Hardware Configuration Definition: User's Guide</i>	SC28-1848
<i>OS/390 Information Roadmap</i>	GC28-1727
<i>OS/390 Information Transformation</i>	GC28-1985
<i>OS/390 Introduction and Release Guide</i>	GC28-1725
<i>OS/390 V1R2.0 JES Commands Summary</i>	GX22-0041
<i>OS/390 Licensed Program Specifications</i>	GC28-1728
<i>OS/390 Printing Softcopy Books</i>	S544-5354
<i>OS/390 Starting Up a Sysplex</i>	GC28-1779
<i>OS/390 Up and Running!</i>	GC28-1726
<i>Planning for the 9032 Model 3 and 9033 Enterprise Systems Connection Director</i>	SA26-6100
<i>Resource Access Control Facility (RACF) Command Language Reference</i>	SC28-0733
<i>S/390 MVS Sysplex Overview – An Introduction to Data Sharing and Parallelism</i>	GC23-1208
<i>S/390 MVS Sysplex Systems Management</i>	GC23-1209
<i>S/390 Sysplex Hardware and Software Migration</i>	GC23-1210
<i>S/390 MVS Sysplex Application Migration</i>	GC23-1211

<i>Table 2 (Page 2 of 2). Related Product Books</i>	
Title	Order Number
<i>S/390 Managing Your Processors</i>	GC38-0452
<i>TSO/E REXX/MVS Users Guide</i>	SC28-1882
<i>TSO/E REXX/MVS Reference</i>	SC28-1883
<i>VSE/SP Unattended Node Support</i>	SC33-6412
<i>VSE/ESA 1.1.0 Unattended Node Support</i>	SC33-6512
<i>VTAM Version 3 Release 3 Network Implementation Guide</i>	SC31-6404
<i>VTAM Version 3 Release 4 Network Implementation Guide</i>	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.

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

Part 1. Understanding CICS Automation

Chapter 1. The Components of Automation

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

- Sophisticated startup and shutdown processes that include:
 - The ability to determine if XRF is used so that CICS Automation can track the active and alternate regions during operation and issue the correct commands during shutdown.
 - The ability for the programmer to define periods of availability to automatically start up and shut down CICS regions. This is referred to as *service periods*.
 - The ability for the programmer to identify external events, such as batch jobs, that work with service periods to determine when to start up or shut down a CICS region. This is referred to as *triggers*.
- Initialization and recovery of interregion and intersystem communications (IRC and ISC) and VTAM application control blocks (ACB).
- Transaction recovery procedures.
- A single-point-of-control that allows you to monitor and control all CICS regions in your organization, local and remote, from one NetView operator session.
- The ability to issue CICS master terminal transactions (CEMT) to CICS regions on remote as well as local domains.
- The ability to “health check” applications running under CICS.

This chapter briefly describes some of the automation components used with SA OS/390 and CICS Automation:

The NetView Automation Table

How CICS Automation is driven. See “The NetView Automation Table” on page 4.

Automation Operators

Where the actions run. See “Automation Operators” on page 4.

The SA OS/390 Control File

Defines which actions to take. See “The SA OS/390 Control File” on page 5.

The SA OS/390 Status File

Tracks actions and maintains status. See “The SA OS/390 Status File” on page 5.

The State/Action Tables

Invoke actions based on event and status relationships. See “The State/Action Tables” on page 6.

The Program-to-program Interface

How you communicate with CICS Automation. See “CICS Automation’s Use of the Program-to-Program Interface” on page 6.

The Operator Interface

How the Operator deals with different environments. See “The Operator Interface” on page 7.

For a complete and more comprehensive list of automation components, refer to the SA OS/390 base documentation.

The NetView Automation Table

Many (but not all) of the automation processes are started from this table. Basically, the automation table is scanned whenever a message is received by NetView to see if actions are required as a result of the message. Review the following example:

```
IF MSGID = 'XYZ123'  
  THEN EXEC( CMD('EVEXZ123 '));
```

This states, “When message **XYZ123** is received, execute **EVEXZ123**.” EVEXZ123 verifies that the region that issued the message is defined in the control file and then takes the automated actions defined for this message.

This is a simple example of a automation table entry. The automation table is a very powerful automation component that enables immediate and consistent responses to messages and enables CICS Automation to account for differences between CICS Version 2, Version 3, and Version 4. As a result, any delay in response to events which may affect integrity or availability is prevented.

CICS Automation provides an extension to the automation table that defines default automation responses to CICS messages. Refer to “CICSMSG1 (EVEMSG01)—Production NetView Initialization Automation Table” on page 247.

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 MVS task. This task design 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 CICS Automation:

CICSMSTR This keyword is used to identify the automation operator that provides message distribution services for CICS 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 CICSOP nn keywords, which are region specific.

CICSOP nn These keywords are used to identify the automation operators that handle messages for specific CICS regions.

CICSCPPI Handles program-to-program interface communications from CICS.

The SA OS/390 Control File

The SA OS/390 control file is a very important piece of the automation process. It contains definitions of the system environment and the resources to be automated. A programmer defines automation policies used throughout the organization by coding them into the control file or by using the SA OS/390 customization dialog (SA OS/390 users only).

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

- Service period entries (when to start up and shut down CICS).
- Group definition entries to start up or shut down groups of CICS subsystems.
- Trigger definition entries to correlate startups and shutdowns with external events.
- Entries that indicate which transactions should be purged during a startup or a shutdown.
- IRC and ISC link monitoring entries.
- CICS application health check entries.

Here is an example of a control file entry:

```
SERVICE CICS1,  
    DAY=(WEEKDAY,0830-1700),  
    DAY=(WEEKEND,DOWN-DOWN)
```

This is the **SERVICE** entry used to define the service periods for the CICS subsystem known as **CICS1**. In this example, CICS starts up at 08:30 every day of the week and shuts down at 1700. It stays down for the weekend.

If needed, the operator interface can be used to redefine a service period for a given day. Also, other selected control file entries can be overridden through the operator interface.

For more information about coding the control file, refer to Chapter 3, “Programming the Control File” on page 41.

The SA OS/390 Status File

The SA OS/390 status file maintains a log of startup and shutdown events, as well as service periods and triggers. It is used, along with the control file, to determine when and how automation is to occur.

The status file is also used to preserve CICS subsystem information during NetView recycles, a function required for CICS 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, the operator defined service periods are merged with those from the control file. Operator definitions take precedence.

“EVESTS - Status file interface” on page 99 describes how data in the status file is retrieved and updated.

The State/Action Tables

These tables are a generalized approach to automating resources based on events, states, transitions between states, and actions taken during state changes.

States can be thought of as the condition of the resource at a given time. The next state is determined by an event, such as a message, and the current state. Actions may be performed in conjunction with the change to the next state. These factors are determined by the State/Action Table matrix.

State/Action Tables are used for problem recovery of entities within a region, such as VTAM ACB failures, storage violations, and short-on-storage conditions. They work independently of service periods and external triggers and are referenced and updated when specific events occur.

State/Action Table definitions are tied into the control file with the PRODUCT= entry. This associates the definitions with a particular subsystem or resource. The State/Action Table can be modified by the programmer to suit your specific needs. Refer to "How to Use the State/Action Tables" on page 24.

CICS Automation's Use of the Program-to-Program Interface

NetView's program-to-program interface provides the ability to communicate between a NetView application and other address spaces on the same host, such as CICS and IMS. CICS Automation uses this program-to-program interface to:

- Initiate, from NetView, the execution of a CICS program.
- Process a response from this CICS program.
- Initiate, from CICS, the execution of a command list or command processor in NetView.
- Process a response from this command list or command processor.

There are CICS Automation program-to-program interface components in CICS as well as in NetView. Part 3, "Planning, Installation, and Customization" on page 149 provides you with step by step procedures that tell you how to install these components so that the interface can be implemented.

You do not need to understand our implementation of the NetView program-to-program interface to use the CICS Automation-provided functions. However, if you are considering using the CICS Automation program-to-program interface code for your own purposes, read Appendix A, "CICS Automation and the Program-to-Program Interface" on page 199.

NetView Components

There is an optional task (EVENTASK), an initialization member for this optional task, and command processors. The initialization member is described in "EVENTASK - NetView PPI initialization member" on page 141. The command processors are described in "EVESNCCI - NetView to CICS Communication Interface" on page 207 and "EVESNRSP - Common Response Handler from CICS" on page 212.

CICS Components

There is a long running program COPC, start and stop functions to start and stop the CICS program-to-program interface component, a subroutine program which is described in “EVESCCCI - CICS to NetView Communication Interface” on page 213, and an initialization member which is described in “EVESPINM - CICS PPI initialization member” on page 139.

Communication Components

An ID (RECEIVERID) is defined at both ends of the program-to-program interface, so CICS Automation knows which NetView to sign on to. This RECEIVERID is contained in the initialization members described above. The VTAM applid is used by CICS to sign on to the program-to-program interface. NetView determines which applid relates to which subsystem from the CICSCNTL control file entry.

The CICS Automation program-to-program interface cannot function if the RECEIVERIDs in the initialization members do not match or if the VTAM applid in the CICSCNTL control file entry does not match the actual VTAM applid. The steps contained in the customization sections describe how to define the RECEIVERID and the VTAM applid.

The Operator Interface

The operator interface is used to monitor your CICS regions and to initiate CICS Automation operator tasks. Many of the operator tasks start pre-defined automation processes, thus allowing consistent automation policies to be maintained. To use the operator interface, enter **CICS** on a NetView panel command line. This displays the CICS Automation Main Menu. Refer to Figure 1:

EVEK0000 CICS AUTOMATION: MAIN MENU		Page: 1 OF 1
(C) 5685-151 Copyright IBM Corp 1991, 1995		Date: 01/16/95
Subsystem, group or domain _____ (? for list)		Time: 11:26
Select an option		
1. Inquire	Display detailed status of CICS subsystem	
2. Start	Start a CICS subsystem, group or domain	
3. Shutdown	Shutdown a CICS subsystem, group or domain	
4. Triggers	Display start and shutdown trigger conditions	
5. Service Periods	Perform service periods functions	
6. Master Terminal	Perform master terminal functions	
7. Monitoring	Perform monitoring functions	
8. Broadcast	Send message to specific CICS subsystem(s)	
9. Support	Provide support functions	
99. Local Functions	Provide access to user defined local functions	
Command ==>		
F1=Help	F2=End	F3=Return
		F6=Roll

Figure 1. The CICS Automation Main Menu

Many functions are region, group, or domain specific. Entering a question mark in the main menu “Subsystem, group, or domain” field displays the CICS Automation Subsystem Selection panel that allows you to list regions, groups, or domains for selection.

To select an option, enter an option number in the main menu “Select an option” field. If you want to use the Status Display Facility, enter **SDF** on the command line.

Option 99 provides a convenient way to access your local applications from the CICS Automation Main Menu panel, thus allowing your CICS operators to work from a single interface.

The CICS Automation panel options are used to:

- **Display and monitor current status**

The Inquire option, listed on the main menu, allows the operator to view detailed information about a CICS region, such as:

- CICS region status
- Current executing domain
- Previous region start time
- VTAM attributes
- Last start type performed.

- **Monitor links and application health**

Another main menu option, Monitoring, allows the operator to work with the link monitoring and health checking functions.

Link monitoring is used to monitor the links between CICS regions and other resources or systems. The operator interface allows the operator to start, stop, recover, or list the status of all links for a given region.

Health checking verifies that all resources required for specific applications are available. It is used to notify the operator when problems are identified. Specific health check routines can be requested at any time with the health check panels. These panels also allow the operator to start and stop automatic health checking.

- **Start up or shut down CICS**

Two options on the main menu (option 2 for start and option 3 for shutdown) allow the operator to start up or shut down a CICS region or a group of CICS regions.

When an operator uses these options, the predefined startup and shutdown actions defined in the control file are used. This includes service periods (the times that the subsystem should be active) and triggers (switches set on or off based on external events). These are validated to verify that the CICS region should be started or shut down. If there is a conflict, the operator is notified and asked to confirm the startup or shutdown request before the procedure is initiated.

The CICS system initialization table (SIT) parameters can also be overridden by the operator.

- **Work with service periods and triggers**

The panels displayed with these options (option 4 for triggers and option 5 for service periods) show the defined service periods for this subsystem or the trigger conditions required for a startup or shutdown of this subsystem. From these panels, you can temporarily change the settings.

For example, if you need to apply maintenance at a specific time, and that time is within a service period window, you can change the setting using the service period panels to initiate a shutdown and a startup as required. This information is stored in the status file.

Records are maintained in the status file for each subsystem. These records contain, among other types of information, a list of triggers. Triggers are “posted” (set off or on) in the status file. The operator interface allows you to override a required trigger condition so that an automated startup or shutdown can occur.

These panels do not allow you to change the permanent service period or trigger requirements defined in the control file. Because the changes are made in the status file, which is dynamically updated, you have the flexibility of changing the programmed parameters for special circumstances, thus taking advantage of automated startups and shutdowns without changing the defined policies.

- **Issue CICS CEMT master terminal transactions**

The Master Terminal option, accessed from the main menu, allows the operator to perform CICS master terminal transactions for the following:

- Terminal control functions
- Task control functions
- Transaction control functions
- Data control functions
- Program control functions
- Inter-CICS connection management
- CICS system information
- Dump control functions
- Transient data queue functions
- Journal control functions
- Automatic installation of terminals
- Trace control functions
- Free-format CEMT requests.

Two advantages of the CEMT interface are:

1. It directs commands from your local NetView to any CICS subsystem, local or remote.
2. It can be used by operators who are not familiar with CEMT syntax.

- **Broadcast messages to users**

Informational messages can be sent to users of CICS regions executing under CICS Automation. Warnings regarding abnormal processing can be transmitted, as well as messages informing users of planned (or unplanned) database and other resource outages. The Broadcast option is listed on the main menu.

- **Work with the program-to-program interface**

You can initialize and de-activate either end of the program-to-program interface (CICS side or NetView side) with the CICS Automation Program-to-Program Interface panel. This panel also shows you the status of each interface. The panel is accessed through the support functions (option 9 from the main menu).

- **Adding local applications to the CICS Automation operator interface**

Option 99, Local Functions, on the CICS Automation Main Menu, enables you to add your local applications to the CICS Automation interface. “Adding Local Applications to the CICS Automation Operator Interface” on page 33 explains how this is done.

Sysplex Considerations

The CICS Automation customization that is currently used with CICS Automation 1.3 can be used unchanged with CICS Automation 1.4.

Version 1 Release 3 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, which 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 CICS Automation, a subsystem name can occur in the 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 CICS Automation, the subsystem is initially brought up on the primary system, and CICS 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.

Chapter 2. How to Implement and Use CICS Automation

In an automated environment, startup, shutdown, and recovery procedures are initiated by events, such as messages, operator requests, and timers. CICS 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 tell you when to use them.

Table 4 (Page 1 of 2). Samples and Source Used for Customization

Member	Purpose
EVECFG	This is the %INCLUDE member that points to EVECFG01, the member that contains definitions for CICS Automation. It is merged into the SA OS/390 control file. See “EVECFG—%INCLUDE Statement for CICS Automation Control File Entries” on page 229.
EVECFG01	Sample control file definitions for a CICS subsystem. See “EVECFG01—Sample CICS Automation Control File Entries” on page 229.
EVETREE	The CICS Automation panel tree definition. See “How to Set Up the Status Display Facility for CICS Automation” on page 14 for a description of this member. “EVETREE—The Status Display Facility Tree Structure” on page 282 shows the sample member.
Status Display Facility panels	Several Status Display Facility panels are included in the CICS Automation package. See “How to Set Up the Status Display Facility for CICS Automation” on page 14 for a description of some of these panels.
EVEPNLS	The CICS Automation panel %INCLUDE member that contains %INCLUDE statements for all of the Status Display Facility panels. “EVEPNLS—The Status Display Facility Panels” on page 283 shows the sample member.
CICSMSG0 (EVEMSG00)	Automation table (%INCLUDEs EVEMEV10 and EVEMCON0). See “CICSMSG0 (EVEMSG00)—Early NetView Initialization Automation Table” on page 244.
CICSMSG1 (EVEMSG01)	Automation table (%INCLUDEs EVEMEV11, EVEMCON1, and EVEMDFH1). See “CICSMSG1 (EVEMSG01)—Production NetView Initialization Automation Table” on page 247.
EVESCMT2	Message exit table for CICS Version 2. This exit captures specific CICS Version 2 messages that are not sent to a console and routes them to the NetView Automation Table so that automation can take place. See “EVESCMT2—Message Exit Table for CICS Version 2” on page 267.
EVESCMT3	Message exit table for CICS Version 3 and higher. This exit captures specific CICS messages that are not sent to a console and routes them to the NetView Automation Table so that automation can take place. See “EVESCMT3—Message Exit Table for CICS Version 3 and Higher” on page 271.
EVEESA01	The CICS Automation State/Action Table used for short on storage recovery. See “EVEESA01—Short-on-Storage State/Action Table” on page 284.
EVEESA02	The CICS Automation State/Action Table used for storage violation recovery. See “EVEESA02—Storage Violation State/Action Table” on page 286.
EVEESA03	The CICS Automation State/Action Table used for autoinstall problem recovery. See “EVEESA03—Autoinstall Problem Recovery State/Action Table” on page 287.
EVEESA04	The CICS Automation State/Action Table used for VTAM ACB problem recovery. See “EVEESA04—VTAM ACB Problem Recovery State/Action Table” on page 288.

Table 4 (Page 2 of 2). Samples and Source Used for Customization

Member	Purpose
EVECMD	CICS Automation specifications that are %INCLUDEd into NetView's Command Model Statements member. See "EVECMD—NetView Command Model Statements Member CICS Automation Entries" on page 277.
EVEDMN	CICS Automation specifications that are %INCLUDEd into DSIDMN. See "EVEDMN—NetView Domain Member CICS Automation Entries" on page 280.
EVEOPF	CICS Automation specifications that are %INCLUDEd into DSIOPF. See "EVEOPF—NetView Operator Definitions CICS Automation Entries" on page 281.
EVESECUR	Security definitions. See "EVESECUR—Security Definitions (Included by EVECMD)" on page 279.
EVEPRFAO	The CICS Automation operator profiles that are added to the NetView profile library. See "EVEPRFAO—CICS Automation Operator Profiles" on page 291
EVECHLTH	A health check sample program (COBOL version). See "EVECHLTH—Health Check Sample Program (COBOL Version)" on page 292
EVECAREA	A copy book layout of the health checking DFHCOMMAREA passed to a user-written health checking application. See "EVECAREA—Layout of the Health Checking DFHCOMMAREA" on page 295
EVENTASK	This is used to initialize program-to-program interface on the NetView side of the interface. See "EVENTASK—PPI Initialization Member (NetView)" on page 296
EVEPINM	This is used to initialize program-to-program interface on the CICS side of the interface. See "EVEPINM—PPI Initialization Member (CICS)" on page 298
Echoplex backend programs	RPG programs, communication files, and command lists are provided for echoplexing (a function of link monitoring) to S/38, AS/400, S/36, and AS/ENTRY systems. This code is placed on the target systems.
Common routines	Command lists and command processors that you can use to write your own automation processes. See Chapter 4, "CICS Automation Routines, Commands, and Definition Members" on page 87.

How to Use Single-Point-of-Control

First, you need to know about gateways and focal point services.

Automation networks can be configured in such a way that messages can be routed between systems. Focal point services, a feature of SA OS/390, is used to define the domains to which messages are forwarded. This allows the operator to monitor systems in a distributed environment. Refer to Figure 2 on page 13.

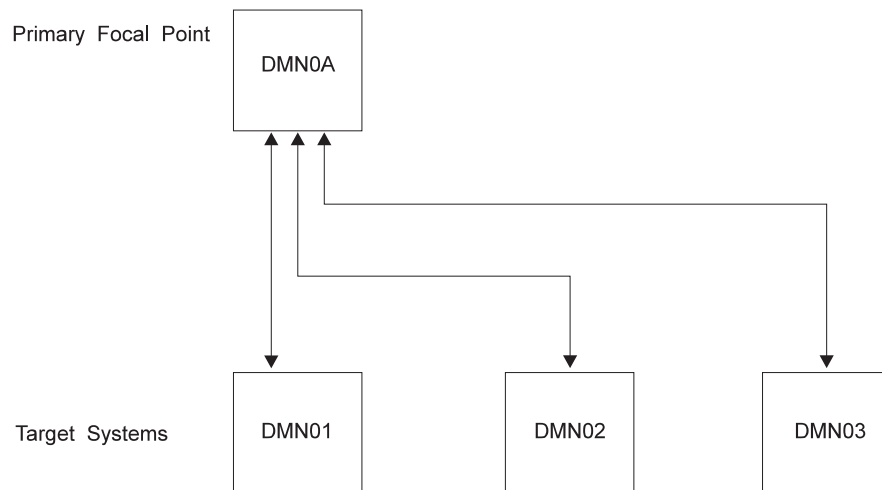


Figure 2. Focal Point Definitions in a Distributed Environment

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 Status Display Facility panels. In the scenario depicted in Figure 2, 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 as well as monitor all of the systems from a single operator session on the focal point system. All functions available on the focal point system CICS Automation operator interface are also available to the target systems; such as inquiries, startups, shutdowns, CEMT transactions, and so on.

This Is How It Works

When a target domain is initialized, and a focal point system is designated for that target system, static information about the CICS 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 CICS subsystems reside, transactions 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 CICS Automation.

Note: If you are using the CICSPOST command processor to post triggers across domains, as described in “Using Triggers in a Distributed Environment” on page 21, gateways must be defined between the “from-to” domains.

This Is Required

Single-point-of-control requires that the subsystem names used for CICS regions be unique across the enterprise. CICS 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 to one and only one NetView domain or the results are unpredictable.

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

How to Set Up the Status Display Facility for CICS Automation

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

The Status Display Facility 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:

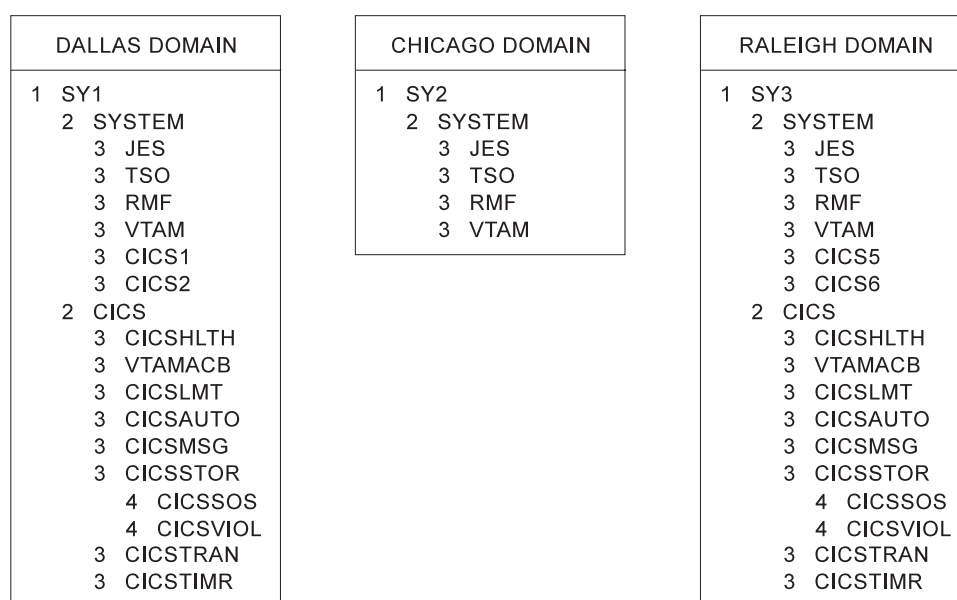


Figure 3. Tree Structures of a Distributed Environment

The three tree structures in this example represent a distributed environment. The files containing the tree structure are located on the actual system for which 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 the Status Display Facility to represent a subsystem. It is a reserved word used to indicate that the resources under this entry are subsystems. CICS, also a reserved word, represents a group of errors that can be generated from CICS Automation. The words listed under the CICS word are the reserved words used to define the group of CICS errors. When you look at the panel in Figure 4 on page 17 you will see the relationship between these words and actual displayed data.

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 a large number of subsystems. This is described in the SA OS/390 base documentation.

The **SDF** control file entry is used with the Status Display Facility, as follows:

SDF *status*

SDF is the control file entry that defines the priority, highlight level, and color definitions for each status. Several keywords were added to this entry to support CICS Automation:

CICSTRAN	CICS Transactions.
CICSHLTH	Health checking.
CICSLMT	Link monitoring.
CICSTIMR	CICS timers.
CICSVIOL	Storage violations (CICS storage).
CICSSOS	Short on storage conditions (CICS storage).
VTAMACB	VTAM ACB errors.
CRITMSG	The default critical messages definition (CICS Critical Message).
CRITMSGA	Messages ending in A (CICS Critical Message).
CRITMSG E	Messages ending in E (CICS Critical Message).
CRITMSGW	Messages ending in W (CICS Critical Message).
CRITMSGI	Messages ending in I (CICS Critical Message).

The color definitions indicate which color to use on the CICS Monitor panel when a message is logged against a specific category, as defined with the control file entry described in “SDF—Status Display Facility Color Definitions” on page 52. CICS Critical Messages have multiple message types. 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 CICS Monitor panel CICS Critical Messages item depends on the message logged with the highest priority. So, if a message ending in A is logged, the CICS Critical Message item will turn to the color defined for those messages (probably red), overriding any other message color.

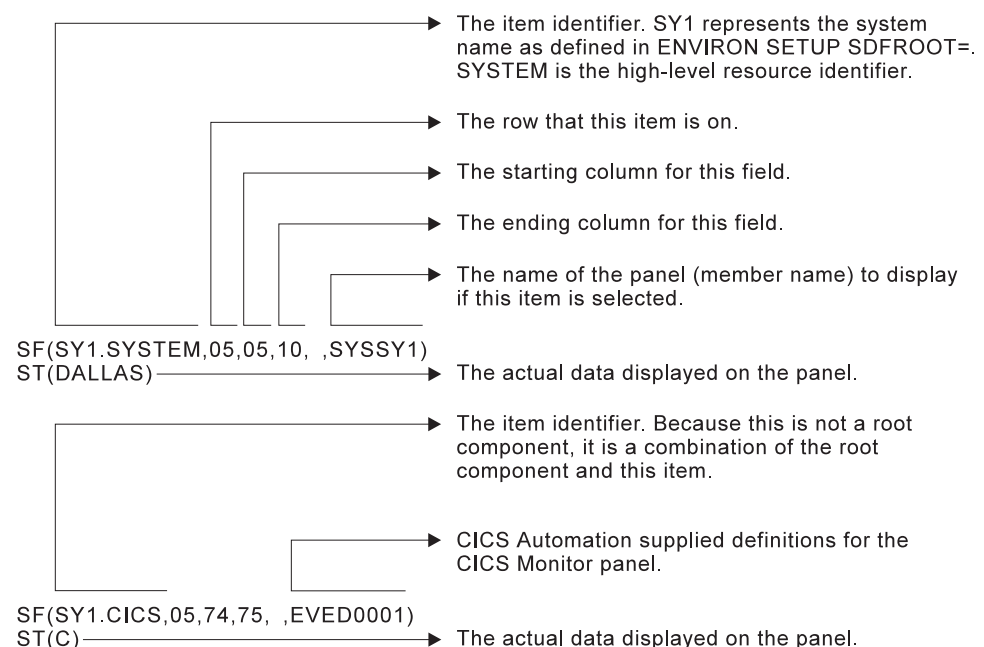
The CICS Automation Status Display Facility Panels

You can use the panel definition keywords to display the distributed environment shown in Figure 3 on page 14 as shown:

SYSTEM		AOC/MVS SUPPORT SYSTEMS							
System	Subsystems	WTORs	Gateways	Spool	MVS Comps	Features			
DALLAS						I C 0			
RALEIGH						C			
CHICAGO						I			
DENVER							0		
PHOENIX	FPIMSEA	IMS401E	AOF05I	JES		I C 0			

This 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, RALEIGH, and PHOENIX, for example, have the letter **C** under the features column. (The CHICAGO and DENVER systems do not have CICS.) This letter represents the “2 CICS” entry in the tree structures. (Do not confuse this with CICS subsystem entries, which are listed under “2 SYSTEM”).

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



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

```
EVED0001          CICS MONITOR PANEL

      CICS Critical Messages

      Health

      VTAM ACB

      Autoinstall

      Link Monitor

      CICS Storage

      CICS Timers

      CICS Transactions

                                                    04/16/95 13:07

====>
PF1=HELP 2=DETAIL 3=END 4=HWD  6=ROLL 7=UP 8=DN          12=TOP
```

Figure 4. The CICS Monitor Panel

This shows several categories in which CICS status is important. If the letter **C** shown on the previous panel was red, then at least one of the items on the CICS Monitor panel will be red. Tab down to the red item and press PF8. This displays the messages logged against that item, as shown in the following:

```
EVEDTIM1          CICS: Timers

      System      Message text
04/16/95 08:00:32 CICSA      'START DENIED BY OTHER EXIT'
04/16/95 08:00:29 CICS06AB   'SUBSYSTEM STATUS IS STOPPED.'
04/15/95 19:00:20 CICSTST    'START DENIED BY OTHER EXIT'
04/15/95 18:00:49 CICSTST1   'TIMED SHUTDOWN PENDING, TRIGGERS NOT SET'

                                                    04/16/95 13:07

====>
PF1=HELP 2=DETAIL 3=RET          6=ROLL 7=UP 8=DN          10=LF 11=RT 12=TOP
```

Note: If the full message is not displayed on the screen, press PF11 to shift to the right.

To see the detail of a message, tab down to that message and press PF2. This displays the following:

```
----- DETAIL STATUS DISPLAY -----
                                         1 OF 4

COMPONENT: CICSA                      SYSTEM   : SY1
COLOR      : RED                      PRIORITY :    200
DATE       : 04/16/95                 TIME     : 08:00:32
REPORTER   : GATA0F06                 NODE     : A0F06
REFERENCE VALUE: CICSA_TI_
'START DENIED BY OTHER EXIT'

===>
PF3=RET 4=FPI 6=ROLL 7=UP 8=DN 9=AST 10=DEL 11=BOT 12=TOP
```

To delete a message, press PF10 from this screen.

Note: If any of the panels show:

1 of x

in the upper right corner of the screen, where x is a number greater than 1, then use PF8 to scroll forward as this indicates that there is more information than what is displayed. PF7 can be used to scroll back.

How to Use CICS Automation for Startups and Shutdowns

There are several things to consider when defining startups and shutdowns:

- 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.
- Service periods determine when to start up or shut down CICS subsystems.
- Triggers can be used to allow startups or shutdowns based on external events.
- There are relationships between all of the above items.
- The operator can override any of these.

Parent/Child Relationships in a Startup or Shutdown

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 do not have a PARENT defined in the SUBSYSTEM control file entry.

The next resources to be started are those resources which have the higher level resource defined as the PARENT, as shown:

```
SUBSYSTEM CICS2,JOB=CICS2,PARENT=VTAM
```

This entry states that CICS2 cannot be started until VTAM is active. Once VTAM is active, CICS Automation will allow a startup for CICS2 **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 or the trigger conditions are met to invoke the startup.

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

Defining the Service Periods

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

```
SERVICE CICS1,DAY=(WEEKDAY,08:30-17:00)
```

which states that CICS1 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 CICS 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 CICS startup or shutdown.

Defining Startup and Shutdown Triggers

This function provides a method of correlating events outside of CICS Automation with CICS 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 is used to define which triggers must be set for a shutdown or startup to occur. See “TRIGGER - Startup and shutdown triggers” on page 85 for more information. Up to 28 unique trigger names can be used for each subsystem.

The CICSPOST command processor

CICSPOST 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. The syntax is as shown:

```
CICSPOST NAME=subsys,FUNCTION={SET|UNSET},EVENT=event  
[,TYPE={STARTUP|SHUTDOWN}]
```

Figure 5. The CICSPOST Command Syntax

where *subsys* is the subsystem name, SET or UNSET indicates whether to set the trigger on or off, *event* is the symbolic name that represents the event, and STARTUP or SHUTDOWN indicates what type of trigger this is. Refer to “CICSPOST - Post an external event” on page 111 for more information.

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'
```

TRIGGERS

This control file entry is used to specify the trigger conditions used to signal the initiation of startup and shutdown processes for specific subsystems. For example:

```
TRIGGER CICS1,STARTUP=(JOBA),  
SHUTDOWN=(JOBB)
```

Posting triggers with user-written routines

To post a trigger from an external event, use the CICSPOST routine, as shown in Figure 5. This is the most commonly used method of posting triggers.

Posting triggers from the automation table

Triggers can be posted through the automation table. For example

```
IF MSGID = 'MSG0123'  
THEN EXEC ( CMD ('CICSPOST NAME=CICS1,FUNCTION=SET,EVENT=JOBA,TYPE=STARTUP'))
```

Posting triggers with control file entries

The following control file entries will post triggers:

EXTCOND

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

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

CMD

This control file entry can be used to set or unset triggers for a specific subsystem when certain messages are received by invoking the CICSPOST command processor.

Posting triggers through the operator interface

The Trigger option, listed on the CICS Automation main menu, allows the operator to:

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

The first panel shown after the Trigger option is selected lists startup and shutdown items. The 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, CICS Automation will wait for the service period window to open to initiate the startup. This also means that 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 regardless of whether or not it is within an open service period window.

These specifications look like this in the control file:

```
TRIGGER CICS1,STARTUP=(JOBA,JOBB,JOBC),
          STARTUP=(SERVICE,JOBD),
          SHUTDOWN=(JOBE)
```

where CICS1 is the subsystem.

To post a trigger from the operator interface, an X is entered in the command field that is to the left of the startup or shutdown item. This displays the next 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 local domain. To establish parent/child relationships in a distributed environment, you can use the CICS Automation startup and shutdown triggers.

For example, let's say that you want to start CICS1, CICS2, and CICS3, which are all on different domains, in the following order:

1. Start up CICS1 on DOMAIN A.
2. When CICS1 on DOMAIN A is active, start up CICS2 on DOMAIN B.
3. When CICS2 on DOMAIN B is active, start up CICS3 on DOMAIN C.

And you want CICS1, CICS2, and CICS3, which are all on different domains, to be shut down in the opposite order. You would code the control file entries as follows:

For subsystem CICS1 on DOMAIN A:

```
TRIGGER CICS1, SHUTDOWN=CICS2DN
CICS1 UP,CMD=(,,'CICSPST NAME=CICS2,FUNCTION=SET,EVENT=CICS1UP,TYPE=STARTUP')
```

For subsystem CICS2 on DOMAIN B:

```
TRIGGER CICS2, STARTUP=CICS1UP
                SHUTDOWN=CICS3DN
CICS2 AUTODOWN,CMD=(,,'CICSPST NAME=CICS1,FUNCTION=SET,EVENT=CICS2DN,TYPE=SHUTDOWN')
CICS2 UP,CMD=(,,'CICSPST NAME=CICS3,FUNCTION=SET,EVENT=CICS2UP,TYPE=STARTUP')
```

For subsystem CICS3 on DOMAIN C:

```
TRIGGER CICS3, STARTUP=CICS2UP
CICS3 AUTODOWN,CMD=(,,'CICSPST NAME=CICS2,FUNCTION=SET,EVENT=CICS3DN,TYPE=SHUTDOWN')
```

Note: This will work only if you have outgoing gateways defined between these systems. For example, the control file on each system must have:

```
GATEWAY domain-name
```

where *domain-name* identifies each NetView domain to which the outbound gateway autotask will log on to. (There must be a GATEWAY entry for each domain that you will post to.) Gateways are described in the SA OS/390 base documentation.

Grouping CICS

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, CICS1). A domain is known by the NetView domain name. A subsystem group name is defined with the CICSGROUP control file entry. The entry looks something like this:

```
CICSGROUP CICSGRP1, MEMBER=CICS1, MEMBER=CICS2, MEMBER=CICS3
```

CICS1, CICS2, and CICS3 are the subsystem names. CICSGRP1 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 steps through each subsystem in the list. Service periods and trigger conditions are checked for each subsystem before the process is initiated. The operator has the option of overriding the service period and trigger conditions.

How to Define Transaction Recovery

Customization of transaction recovery consists of:

- Identifying the transactions that will have recovery automation.
- Identifying the error threshold level when a recovery should take place.
- Identifying specific abend codes when you want recovery procedures to take place (there are probably several that you would want to ignore).
- Specifying the recovery procedure, which usually consists of invoking a command, a routine, and/or sending notification 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. Refer to the following example:

```
RECOVERY CICS1.TRAN,AUTO=N
RECOVERY CICS1.TRAN.PAYR,AUTO=Y
RECOVERY CICS1.TRAN.DBTS,AUTO=Y
RECOVERY CICS1.TRAN.BLNG,AUTO=Y
```

This control file entry states that recovery automation is to occur on CICS1 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 CICS1 except for the three listed below that entry.

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 CICS1.TRAN,CRIT=(3,01:00),FREQ=(3,05:00),INFR=(3,30:00)
THRESHOLD CICS1.TRAN.PAYR,CRIT=(2,01:00),FREQ=(2,05:00),INFR=(2,30:00)
```

When coded this way, recovery will take place for all transactions except PAYR if 3 abends occur within 1 hour. This is the generic or default definition. For PAYR, recovery actions will take place if 2 abends occur within 1 hour.

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

ABCODETRAN

Use the ABCODETRAN control file entry to specify which abend codes to take action on. This entry can also be coded generically or by specific transactions. Refer to the following:

```
ABCODETRAN CICS1.TRAN,
CODE=(PAYR,AEI0,,INCLUDE),
CODE=(PAYR,AKC3,,EXCLUDE),
CODE=(*,*,*,INCLUDE)
```

When coded this way, recovery actions will take place for PAYR on CICS1 when the transaction abend code is AEI0, but recovery will not take place if the transaction abend code is AKC3. (INCLUDE indicates that action should take place. EXCLUDE indicates that action should not take place.) The last item, CODE=(*,*,*,INCLUDE), states that recovery is to take place on all other transactions and transaction abend codes.

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

RCVRTRAN

Use the RCVRTRAN control file entry to define the specific recovery actions. The entry can be coded generically or by specific transaction, as shown:

```
RCVRTRAN CICST1.TRAN,  
      CMD=(,,'MSG OP1,TRAN &EHKCFGV1 FAILED')
```

```
RCVRTRAN CICST1.TRAN.PAYR,  
      CMD=(,,'MSG OP1,TRAN &EHKCFGV1 FAILED')
```

The first recovery action definition will take place for all transactions except PAYR. The second recovery action definition will take place for PAYR.

For more information, refer to the following control file entry descriptions:

- “Automation Flags” on page 48 (see the RECOVERY flag definition).
- “THRESHOLDS—Error Thresholds” on page 58.
- “ABCODETRAN - Transaction abend recovery” on page 62.
- “RCVRTRAN - Transaction recovery” on page 81.

How to Use the State/Action Tables

A major portion of automation falls into the area of event driven processing. The occurrence of an *event* (error message, good message, operator request,...) triggers the execution of a specific *action* (start up, shut down, execute a command list, issue a message,...). As an application is automated and its events are analyzed to determine what actions to execute, it becomes apparent that the event/action relationship can be grouped into two categories:

1. Events in which the action to execute is independent of any other event.
2. Events in which the action to execute depends on a number of events or a sequence of events.

In the first case, the action is determined by the event. It does not matter what events have occurred before. The action executed is dictated by the single event, and no other information is required. This type of event is easily processed by the automation table.

In the other more complicated case, the action to execute can only be determined by analyzing more than one event. The order in which the events occur may also be important. This type of event can be handled by a State/Action Table.

How It Works

Events occur at a moment in time. In between the events, a CICS subsystem can be thought of as existing in a particular *state*, such as active, inactive, starting, stopping, recovering, and so on. It is at the occurrence of an event that the subsystem may change states. Also connected with the occurrence of events are the actions to perform. The State/Action Table puts these three pieces—the events, the states, and the actions—together and defines the relationships and interactions among them.

A State/Action Table is a two dimensional matrix. The columns are the states in which the subsystem can exist. The rows are the events that can occur. An entry in the table contains the *new* state of the subsystem and identifies the action to perform. A *current state variable* keeps track of the *current* state of the subsystem.

When an event occurs, an entry in the table is referenced. The location of the entry in the table is determined by using the current state variable value to identify the column and using the event to locate the row. The entry contains the new state and the action to be performed. The current state variable is set to the new state and the action is performed by calling an appropriate action routine. By identifying an action to be performed with each change of state, we identify:

- When to execute an action.
- What action to execute.

Refer to Figure 6:

/*****				
/* STATES:	NORMAL	SINGLE	DOUBLE	*/
/*		HALTED	HALTED	*/
/* STATE VALUES:	0	1	2	*/

EVENT=DFH0506	EVEES101/1	NA	NA	
EVENT=DFH0507	NOP	EVEES102/0	NA	
EVENT=DFHSM0100I	EVEES101/1	/2	NA	
EVENT=DFHSM0101I	EVEES101/1	/2	NA	
EVENT=DFHSM0106I	NOP	EVEES102/0	/1	
EVENT=DFHSM0107I	NOP	EVEES102/0	/1	
EVENT=CICSDN	NOP	EVEES103/0	EVEES103/0	
EVENT=CICSINIT	NOP	EVEES103/0	EVEES103/0	

Figure 6. Short on Storage State/Action Table Example

For example, assume we have a CICS subsystem that is active. There is a state (a column) defined in the State/Action Table for this subsystem which represents this condition. Let's call it the subsystem's NORMAL state. The subsystem's current state variable value is NORMAL (or 0).

Now, assume message DFH0506 occurs (CICS is under stress, short-on-storage). The message is an event. A row in the table defines this event. Indexing into the table using the message event (row) and the current state variable (column), the resulting table entry indicates that the new state should be SINGLE HALTED (or 1) and the action is to execute the command list EVEES101. EVEES101 schedules a short-on-storage timer to start EVEE104 if the short-on-storage condition does not correct itself within a specific time frame. EVEE104, if triggered, alerts the operator, updates the Status Display Facility, and takes an MVS dump of the CICS subsystem.

If message DFH0507 occurs (CICS is no longer short-on-storage) and the state is SINGLE HALTED, we index into the table using the event (row) and state (column), and determine that the action to take is to execute EVEES102 and change the state to NORMAL. EVEES102 cancels the short-on-storage timer and removes any Status Display Facility messages associated with this error occurrence.

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. 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. (See “EVEEY00S - Common state handler for State/Action Tables” on page 120.)
- Default State/Action Tables for several areas. These are shown in the following sections:
 - “EVEESA01—Short-on-Storage State/Action Table” on page 284.
 - “EVEESA02—Storage Violation State/Action Table” on page 286.
 - “EVEESA03—Autoinstall Problem Recovery State/Action Table” on page 287.
 - “EVEESA04—VTAM ACB Problem Recovery State/Action Table” on page 288.
- Common routines to be used by the action routines.
- Action routines for several areas.
- Entries in the control file.

Figure 6 on page 25 shows the portion of the State/Action Table that contains the rows (events) and columns (states) that determine actions. Figure 7 shows the State/Action Table keywords in this table.

PRODUCT=CICS				
AREA=SOS				
STATE=0				
/*****				
/* STATES:	NORMAL	SINGLE	DOUBLE	*/
/*		HALTED	HALTED	*/
/* STATE VALUES:	0	1	2	*/
/*****				
EVENT=DFH0506	EVEES101/1	NA	NA	
EVENT=DFH0507	NOP	EVEES102/0	NA	
EVENT=DFHSM0100I	EVEES101/1	/2	NA	
EVENT=DFHSM0101I	EVEES101/1	/2	NA	
EVENT=DFHSM0106I	NOP	EVEES102/0	/1	
EVENT=DFHSM0107I	NOP	EVEES102/0	/1	
EVENT=CICSDN	NOP	EVEES103/0	EVEES103/0	
EVENT=CICSINIT	NOP	EVEES103/0	EVEES103/0	

Figure 7. State/Action Table Keywords

PRODUCT is used to define the product category for this State/Action Table, AREA defines the area of events that this State/Action Table works with, and STATE contains the initial state value (optional), normally 0.

State/Action Table Control File Entries

PRODUCT *product*,**SUBSYS**=(*subsys*,**AREA**=*area*)

Sets of State/Action Tables are grouped by product. In this case, the product is CICS. 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 (it's the same AREA name specified in the PRODUCT entry), and *function* is a type of State/Action Table, such as short-on-storage (SOS), 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  CICS,SUBSYS=(CICS1,AREA=CICS),
          SUBSYS=(CICS2,AREA=CICS),
          SUBSYS=(CICS3,AREA=TEST)
AREA      CICS,SOS=EVEESA01,
          VIOL=EVEESA02,
          AUTO=EVEESA03,
          VTAMACB=EVEESA04
AREA      TEST,SOS=TESTSA01,
          VIOL=TESTSA02,
          AUTO=TESTSA03,
          VTAMACB=TESTSA04
```

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

The AREA control file entry provides the names of the State/Action Tables for each area. The AREAs are:

SOS The State/Action Table used for short-on-storage recovery.
VIOL The State/Action Table used for storage violation recovery.
AUTO The State/Action Table used for autoinstall problem recovery.
VTAMACB The State/Action Table used for VTAM ACB problem recovery.

In our example, the CICS3 AREA= keyword for the PRODUCT control file entry specifies TEST. The AREA control file entry for TEST has a different set of tables defined to it.

Refer to "EVEEY00S - Common state handler for State/Action Tables" on page 120.

How to Use Health Checking

Health checking allows you to execute programs that check the health of an application running under CICS. CICS Automation initiates a health check program by sending a request across the program-to-program interface to the target CICS. CICS receives the request, processes the program, and sends the results back to NetView.

Note: Sample health check code is provided with CICS Automation. This code can be modified and used to check the availability of critical resources, such as DB2 or IMS. See “EVECHLTH—Health Check Sample Program (COBOL Version)” on page 292.

Setting Up Health Check Programs

Health checking is set up in the following manner:

Step 1: The health check program is written

The actual health check program is executed on the target CICS. According to the health check protocol, one of the following is returned:

- An acknowledgment (ACK) stating that the program completed successfully.
- A negative acknowledgment (NACK) stating that the program did not complete successfully. If a NACK response is given, data can be passed back to NetView describing the error condition.

The ACK or NACK are returned through the use of a DFHCOMMAREA. The user-written health check program is linked to by an CICS Automation health check program, which passes the 104-byte DFHCOMMAREA. The first 4 bytes are reserved for the characters ACK or NACK. The last 100 bytes can be used for a NACK message if the user-written program encounters an error. Refer to “EVECAAREA—Layout of the Health Checking DFHCOMMAREA” on page 295 for the format of the DFHCOMMAREA, and “EVECHLTH—Health Check Sample Program (COBOL Version)” on page 292 for an example of how to use the DFHCOMMAREA to return the response.

Step 2: The health check programs are defined to CICS

Use either the *CICS/MVS Resource Definition (Online)* or the *CICS/ESA Resource Definition (Online)* publication to do this.

Step 3: The health check program is defined to CICS Automation

The health check program is defined with the HEALTHCHK control file entry. The format looks something like this:

```
HEALTHCHK CICS1,FUNCTION=(0,PAYCHECK,00:02:00,15,, 'Check payroll')
```

Up to 10 health check programs can be defined for each CICS subsystem. Zero (0) is the first one defined. One (1) would be the second one, two (2) the third one, and so on. The program name, in this case PAYCHECK, is used to identify the specific health check program. You will also indicate how often you want the program executed (every two minutes) and how long you want to wait for a response before sending operator notification (fifteen seconds). The description (for example 'Check payroll') is used on the CICS Automation Health Checking panel.

Using the Health Check Operator Interface

Health check programs are executed automatically at timed intervals. When CICS Automation receives an abnormal response from CICS (NACK) or when a timeout occurs, CICS Automation sends a notification. CICS Automation panels are used to manually work with health checking.

Select option 2, Health checking, from the CICS Automation Monitoring panel to display the following:

EVEKM200		CICS Automation: Health Checking		Page: 1 of 1	
Subsystem CICS2		(? for list)		Date: 01/09/95	
				Time: 11:34	
Select a command:		1. Start	3. Suspend	5. Detail	
		2. Stop	4. Resume	6. Immed check	
CMD	Program	Description	Status	Date	Last Status Check Time Response
—	HCPAY1	CHECK PAYROLL DATA BASE	ACTIVE	01/09/95	10:32:00 ABNORMAL
—	HCEDI5	ACCESS TO EDITOR	INACTIV	01/06/95	09:00:00 NORMAL
—	HCFULL	95% FULL CONDITION	ACTIVE	01/09/95	11:30:05 NORMAL
—	HCACTV	95% ACTIVE CONDITION	ACTIVE	01/09/95	09:09:59 NORMAL
—	HCAREC	ACCOUNTS RECEIVABLE	ACTIVE	01/09/95	11:33:00 ABNORMAL
—	HCTERMA	95% TERMINALS ACTIVE	INACTIV	01/06/95	09:00:00 NORMAL
—	HCOUT1	ACCESS TO OUTMAIL FILE	ACTIVE	01/09/95	11:33:10 ABNORMAL
—	HCIN1	ACCESS TO INMAIL FILE	INACTIV	01/06/95	09:10:00 ABNORMAL
—	HCPAY2	PAYROLL SUBMIT	ACTIVE	01/09/95	08:32:55 NORMAL
—	HCTERM2	REMOTE TERMINAL PROGRAM	INACTIV	01/03/95	08:00:00 NORMAL
Command====>					
F1=Help		F2=End	F3=Return	F4=CICS menu	F5=Refresh F6=Roll

Figure 8. Health Checking panel

This panel lists the health check programs for a particular subsystem. The list includes the:

- Program name
- Description
- Status (ACTIVE, INACTIV, or SUSPEND) which shows whether automatic execution is active
- A time stamp of the last time the program ran
- Response (NORMAL or ABNORMAL).

From this panel, you can:

- Initiate automation of a program (Start)
- Stop automation of a program (Stop)
- Suspend automatic execution of a program (Suspend)
- Resume automation execution of a program (Resume)
- Show the details of a health check program (Detail)
- Immediately execute a program (Immed check).

How to Use Link Monitoring

CICS communicates with other systems or subsystems using either intersystem communication (ISC) or interregion communication (IRC). In either ISC or IRC, communication between different systems or subsystems takes place across predefined sessions. Sessions are logical links that are allocated whenever there is a need to communicate. Links are defined to CICS using the CONNECTION definition.

Defining the Links to CICS Automation

A four character ID is used to identify the CONNECTION name. CICS Automation has a CONNECTION control file entry that is used to define link monitoring characteristics for each link. This control file entry uses the CICS defined CONNECTION name to identify the link itself. A sample link monitoring control file entry looks like this:

```
CONNECTION CICS1.LNK1,  
          DESC='IRC link to CICS2',  
          DAY=(WEEKDAY,0830-1700),  
          TYPE=CICS  
          ECHO=(ECHO)
```

This entry states that the link known to CICS1 as LNK1 is to be monitored Monday through Friday from 8:30 in the morning until 5:00 in the afternoon. The system at the other end of the link is another CICS subsystem (TYPE=CICS). The ECHO= entry indicates that we want a response from the other CICS subsystem and that the transaction named ECHO is to be used to request this response. If this entry were not included in the definition, then we would only check that link LNK1 is active on CICS1.

What Is Echoplexing

Basic link monitoring ensures that VTAM connections are acquired and available but does not detect problems at the other end of the link. CICS Automation provides the ability to verify the other end of the link by sending data across it and waiting for a response. This is referred to as *echoplexing*. You can echoplex to any system or subsystem as long as the:

1. Link is defined on the primary CICS region
2. Type of connection is either multiregion operation (MRO), LU6.1 or LU6.2
3. Target system is either CICS, or IMS

SA OS/390 and CICS Automation are not required on the target systems. However, the target systems or regions require access to CICS Automation code. This code is supplied as samples and requires modification for your specific environment. CICS Automation uses the IMS /TEST function to get a response from IMS and no additional programming is required.

A CICS target subsystem requires access to EVESYCB7 (a CICS Automation program) and the definitions required for this program are made as shown:

```
DEFINE TRANSACTION(ECHO) PROGRAM(EVESYCB7)  
DEFINE PROGRAM(EVESYCB7) LANGUAGE(ASSEMBLER)
```

Note: The default name for the transaction is ECHO. This can be changed as long as the transaction definition on the target system matches the transaction name on the primary system that is identified with the ECHO= keyword.

How Security Checking Works

There are two different implementations of CICS Automation security. One implementation uses NetView based security techniques, and limits, at a functional level, operator actions in the NetView environment. This functional level security also restricts what functions an operator can invoke in CICS.

The second implementation utilizes CICS supplied security to restrict which operators can access defined resources within a CICS environment. This level of security is only available in the CICS Version 4 or higher environment.

Note: In order to perform any of the basic functions of the CICS Automation Feature, like displaying subsystem information, an operator must be authorized to use the ACF command.

Security Checking Using NetView

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

CICS1	KEYCLASS 1	<u>Operators with class 1 can work with CICS1.</u>
INQUIRY	VALCLASS 1	<u>Operators with class 1 can invoke an inquiry on CICS1.</u>
STARTUP	VALCLASS 2	<u>Operators with class 1 and 2 can invoke a startup on CICS1.</u>
SHUTDOWN	VALCLASS 2	<u>Operators with class 1 and 2 can invoke a shutdown on CICS1.</u>
TRIGGERS	VALCLASS 2	<u>Operators with class 1 and 2 can work with triggers on CICS1.</u>
SERVICEP	VALCLASS 2	<u>Operators with class 1 and 2 can work with service periods on CICS1.</u>
CEMT	VALCLASS 2	<u>Operators with class 1 and 2 can issue or set CEMTs on CICS1.</u>
LMT	VALCLASS 2	<u>Operators with class 1 and 2 can work with link monitoring on CICS1.</u>
HEALTH	VALCLASS 2	<u>Operators with class 1 and 2 can work with health checking on CICS1.</u>
BROADCAST	VALCLASS 2	<u>Operators with class 1 and 2 can broadcast to CICS1.</u>
SUPPORT	VALCLASS 2	<u>Operators with class 1 and 2 can work with support function on CICS1.</u>
EVEPPICT	VALCLASS 2	<u>Operators with class 1 and 2 can start and stop the PPI on CICS1.</u>
=OTHER	VALCLASS 3	<u>Operators with class 1 and 3 can perform any other function on CICS1 not already defined in this group of entries (default).</u>
=OTHER	KEYCLASS 9	<u>Operators with class 9 can work with any other CICS.</u>
=OTHER	VALCLASS 9	<u>Operators with class 9 can perform any function on any other CICS.</u>

Entries are grouped by subsystem, group, or domain, with the subsystem, group, or domain listed as the first item in the group (KEYCLASS). All function items (VALCLASS) following a KEYCLASS item are associated with that KEYCLASS until another KEYCLASS item is encountered.

Note: If you change any of the default KEYCLASS and VALCLASS values shipped in EVESECUR, then you must edit EVEPRFAO and add the new values to the OPCLASS statement. This will ensure that the AUTOTASKs which will be controlling various CICS regions and functions are authorized to do so. If this is not done, message EVE004I, plus EVE551E or EVE554E, may be received at STARTUP or SHUTDOWN. For example, if the AUTOTASK for a CICS is not authorized for health, then these messages will be received at shutdown:

- EVE004I SECURITY CHECK - OPID= CICSOP, KEYVALUE= HEALTH
- EVE551E UNEXPECTED RETURN CODE (RC=S) FROM EVEEMHCD IN EVEET020

About =OTHER

=OTHER is used as the default entry and can be used for either KEYCLASS or VALCLASS.

You can call the CICS Automation security checking program for your own automation processes. Refer to “CICSSEC - Invoke security checking” on page 96.

Special considerations for single-point-of-control

The single-point-of-control operators are security checked before attempting to perform functions on remote subsystems. Because of this, you need to define the remote subsystems in the DSICMD member located on the single-point-of-control domain.

Security Checking Using CICS

CICS Automation security checking using CICS facilities requires non-terminal transaction security checking, a function available on CICS Version 4 Release 1 or higher.

The security check works by using the NetView operator ID that invoked the CICS Automation function. When the function to be performed is invoked in the NetView environment, the invoking operator ID is passed to the CICS system on which the action will be taken. The appropriate transaction or function is invoked, and the NetView operator ID is used in all CICS security checks. This enables a more granular security check using CICS supplied security is than is capable in the NetView environment.

To use this security, you must:

- Have CICS Version 4 Release 1 or higher
- Define all NetView operators which will invoke CICS functions defined to RACF (or your SAF compliant security system). This will include:

Regular NetView operators

NetView autotasks which perform CICS related actions. These autotasks will include those autotasks specifically defined for CICS Automation use and may include the autotasks which process shutdown functions or resynchronization functions.

- Define RACF surrogate authorization for the CICS.
- Connect the NetView operators to the CICS resources in which they will need to access, such as transactions, programs and files. This connection is done through your SAF Security manager (such as RACF).
- Enable the security by modifying the EVESPINM member and specifying USERID=YES to enable extended support.

- Enable non-terminal transaction security in CICS by modifying the CICS SIT to specify XTRAN=YES and XUSER=YES. Additional CICS definitions may require similar modification, such as PLTPIUSER.

Adding Local Applications to the CICS Automation Operator Interface

Option 99, Local Functions, from the CICS Automation main menu, provides you with a way to add your local applications to the CICS Automation interface.

To do this, write a module named EVEEU000 using the usage notes described below. This is the module that 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 CICS Automation. For your panel to be logically consistent with the CICS Automation interface, incorporate these functions.

Programming notes:

1. To exit CICS 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
    EVE_PF2 = 'YES'
    'GLOBALV PUTT EVE_PF2'
    EXIT 0
END
```

and

```
WHEN VIEWAID = 'PF4' | VIEWAID = 'PF16' THEN
DO
    EVE_PF4 = 'YES'
    'GLOBALV PUTT EVE_PF4'
    EXIT 0
END
```

2. 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 EVE_PF2'
IF EVE_PF2 = 'YES' THEN
DO
    EXIT 0
END
```

and

```
'GLOBALV GETT EVE_PF4'
IF EVE_PF4 = 'YES' THEN
DO
    EXIT 0
END
```

3. To handle a fast-path command entered on your panel:

- a. Add the following to the beginning of the program:

```
'SIGNAL ON HALT'
```

- b. Add the following routine into the program:

```
HALT:
  EVE_PF2 = 'YES'
  'GLOBALV PUTT EVE_PF2'
  EXIT 0
```

- c. Add the following code after displaying your panel and accepting input:

```
WHEN VIEWAID = 'ENTER' & CMD ~= '' THEN
  DO
    IF SUBSTR(CMD,1,1) = '=' THEN
      DO
        PARSE VAR CMD '=' REST
        CMD = 'EVEE0000 ' || REST
      END
      'CMD HIGH 'CMD
    END
  END
```

Note: In this code, CMD is the command line on the NetView panel.

4. If you code a menu panel, add the following code to check for fast-path when your program is entered:

```
'GLOBALV GETT EVE_SELECTION'
IF EVE_SELECTION ~= ''
  DO
    PARSE EVE_SELECTION MYSELECTION '.' EVE_SELECTION
    'GLOBALV PUTT EVE_SELECTION'
  END
```

5. On entry, or returning from a called program, to get the CICS subsystem name (if the previous program had a valid name and saved it) code the following:

```
'GLOBALV GETT EVESELNM'
MYNAME = EVESELNM
```

6. Always validate a new CICS name before storing it for other programs to use. Following is an example of validation:

```
'CICSQRY REQ=VALIDATE,TYPE=CICS,NAME='MYNAME
IF RC ~= 0
  DO
    write your error message
  END
ELSE
  EVESELNM=MYNAME
  'GLOBALV PUTT EVESELNM'
```

Using Linemode Functions

Linemode functions allow the operator or user-written routines to access the following CICS Automation Automation functions without using the CICS Automation panels:

- Health checking
- CICS startup
- CICS shutdown
- SIT override
- Link monitoring
- Message options
- Trace options
- CICSPOST
- CEMTPPI

In general, the linemode routines make possible the extension of automation from user-written routines. The user-written routine issues the linemode command during NetView initialization or at a specific time or day. A message and return code is given to the calling routine to verify that the requested operation was successful.

Health Checking

Linemode health checking makes it possible to manipulate health check routines from a user-written command. A health-check program is a user-written routine which executes periodically to ensure that a critical application is capable of supporting its users. The actions supported include suspending and resuming the health check program. Other actions are supported.

CICS Startup and Shutdown

Linemode Startup and shutdown commands make it possible to manipulate CICS regions from a user-written command. Much of the flexibility that is built into the CICS Automation panels is supported in the linemode command. Many exception conditions like service period and trigger checks can be bypassed. The ability the perform startup and shutdown processing from a user-written program allows for greater flexibility regarding conditional execution of CICS regions.

SIT Override

Linemode SIT overrides allows a user-written routine the capability to set the SIT overrides, which can then be used by CICS Automation to control the startup of the CICS. A typical use of this linemode command will be to enable automation to perform cold startups on a given day of the week. For example, using SA OS/390 timer facilities, a user could set a timer to set the overrides to cold-start every Monday morning. Then, using CICS Automation service periods, the CICS system could be recycled, and a cold start would be performed.

Link Monitoring

Linemode link monitoring provides support for the link monitoring functions of CICS Automation. Support for most of the link monitoring capabilities are provided, excluded are system news update, service period update and recover all links functions.

Message options

Linemode message options enables a user-written routine to change the message header options which display on the operator panel. Typical use of this command is during NetView initialization, when a user-written routine would set the domain-wide defaults.

Trace Options

Linemode trace options enables a user-written routine to change the CICS Automation trace options. Trace is an internal CICS Automation tool which is used for problem determination. Typical use of this command is during NetView initialization.

CICSPOST

You can use the CICSPOST routine to set trigger conditions in the status file.

CEMTPPI

CEMTPPI allows you to code a CEMT command:

1. In your own automation routines.
2. In the automation table.
3. In the control file using a control file entry that allows you to define specific commands, such as SHUTNORM, STARTUP, RCVRSOS, and so on.

It accepts CEMT input as data, sends it across the program-to-program interface, and sends a response back to the originating task.

How to Implement Remote Site Recovery for VSAM RLS (CICS TS Function Only)

CICS TS provides support for remote site recovery where VSAM data sets are used in RLS mode at the primary site. Using this RLS support for remote recovery, you can switch over to the remote site without suffering indeterminate or unreported loss of data integrity.

To invoke CICS RLS support for off-site recovery, you must start CICS systems manually using start type `START=AUTO` and the system initialization parameter `OFFSITE=YES`. You may choose these options via the CICS Automation Operator Interface startup panel.

With RLS recovery in operation during an emergency restart, CICS prevents any data sets to be accessed in RLS mode until CICS has completed all outstanding RLS recovery work and it has received a 'GO' response to `WTOR DFHFC0575`.

The operator should reply 'GO' to the message only when all the CICS regions being restarted with `OFFSITE=YES` have issued message `DFHFC0575` indicating that they have completed their RLS recovery.

CICS TS provides a sample REXX exec `DFH$OFAR` to be used to automatically reply 'GO' to the `WTOR` for each participating CICS system, when appropriate.

To be able to use the CICS TS provided sample REXX exec `DFH$OFAR`, you will need to copy it from the CICS TS `DFHSAMP` library into a `DSICLD` concatenated library. Refer to CICS TS documentation for more information.

`DFH$OFAR` requires that a unique control file (a sequential dataset) be defined containing all the participating CICS systems. This control file must be accessible from any participating MVS image within the sysplex. Please refer to the prolog in the REXX exec `DFH$OFAR` for more detailed information.

CICS Automation provides the NetView Automation Table entries required to drive the CICS TS provided REXX exec `DFH$OFAR`. You must merge these entries into your own NetView Automation Table to be able to use this function.

Part 2. Programming Information

This part describes CICS Automation common routines which request information or perform tasks associated with CICS 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.

Chapter 3. Programming the Control File

This section describes the format of CICS 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. 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 CICS10AA.TRAN.SAMPLE1,CRIT=...
```

uses underscores as separators if you use Release 1:

```
THRESHOLDS CICS10AA_TRAN_SAMPLE1,CRIT=...
```

Upgrading from CICS Automation 1.3 to CICS Automation 1.4

The CICS Automation customization that is currently used with CICS Automation 1.3 can be used unchanged with CICS Automation 1.4.

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 *System Automation for OS/390 Customization*.
- Likewise, CICS unique customization, such as CICSNTL, SERVICE, TRIGGER, is needed wherever ARM can restart the subsystem. Including the same member that contains only the CICS unique information in each ACF file can simplify this step.

Special Considerations when Coding Control File Entries

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

Restrictions

CICS uses the NetView KEYCLASS statement for security checking of CICS subsystem names. NetView limits the length of a keyword in the KEYCLASS statement to 8 characters. This also limits the length of a CICS subsystem name to 8 characters, although SA OS/390 supports 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 46.

%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 CICS 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 CICS Automation capabilities.

Those entries unique to CICS Automation

CICS Automation provides additional entries required for CICS 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 CICS Automation-unique control file entries used by CICS 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 CICS Automation are supported with the SA OS/390 customization dialogs. Consider the following when deciding whether to use the customization dialogs for CICS 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 CICS 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 5 on page 43.

<i>Table 5. Control File Configuration Data</i>	
Configuration data	What it does
SUBSYSTEM CICS,JOB=CICSONE, PARENT=VTAM	Defines the symbolic CICS name, the job that starts this CICS, and the parent subsystem.
CICSNTL CICS1,APPLID=VTAMAPPL,MAJNODE=VTAM1	Defines the VTAM name for this CICS and the VTAM node.
SERVICE CICS1,DAY=(WEEKDAY,0800-1700)	Defines the service periods in which this CICS 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 9.

```

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

```

Figure 9. 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 9 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 CICS specific customization be used. For example, in Figure 9 CICSALL 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: 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.
- A series of three periods (...) 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 variables; substitute your own value for them.
- Uppercase items must be entered exactly as shown.
- Parentheses must be entered as shown.
- Where operands can be abbreviated, the abbreviations are shown in capital letters. For example, ALL can be entered as A or ALL.
- Where brackets ([]) are nested, you must include commas to denote the absence of the required positional operands. For example, enter XYZ A,,C to specify C only for the following command syntax:

XYZ [A[,B[,C]]]

About the commas

Commas are used as delimiters between parameters. The last parameter does not require a comma after it. Because of this, we place the comma in front of a parameter to show that if you add this parameter, you need a comma. However, the comma actually goes after the preceding parameter and needs to be on the same line as that parameter.

Base SA OS/390 Control File Entries

This section describes base SA OS/390 control file entries that are used for defining CICS Automation. The following table lists those entries, provides a brief description, and indicates whether or not the SA OS/390 customization dialog is used.

Entry	Description	SA OS/390 build
"%INCLUDE—Include Additional Members in Control File" on page 46.	Use this statement to include additional members when the control file is loaded.	Editor
"ACORESTART—NetView Restart Commands" on page 47.	Use these control file entries to call the CICSRSYC command when NetView is recycled. This provides resynchronization of CICS regions already active.	Dialog
"Automation Flags" on page 48. <ul style="list-style-type: none">• INITSTART• RESTART• RECOVERY	Automation flags enable or disable automation for specific types of events.	Dialog
"AUTOOPS—Automation Operators" on page 50.	Defines the NetView operator IDs that function as automation operators.	Dialog
"SDF—Status Display Facility Color Definitions" on page 52.	Describes the keywords that were added to the SA OS/390 Status Display Facility control file entry to support the Status Display Facility CICS Monitor panel.	Dialog
"ENVIRON SETUP—CICS Initialization Support" on page 53.	This entry is used to define the basic SA OS/390 operating characteristics.	Dialog
"ENVIRON TIMEOUT—CICS CEMT Support" on page 54.	This entry is used to define the amount of time to wait for specific replies before issuing error messages.	Dialog
"Shutdown Control File Entries" on page 55.	This entry defines installation dependent commands that are executed when a shutdown is initiated.	Dialog
"STARTUP—Startup Commands" on page 56.	This entry defines installation dependent commands that are executed when a startup is initiated.	Dialog
"SUBSYSTEM—Subsystem Definition" on page 57.	This entry defines the base SA OS/390 information required for the definition of the subsystem.	Dialog
"THRESHOLDS—Error Thresholds" on page 58.	Defines threshold levels as infrequent, frequent, or critical.	Dialog/ Editor

%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
```

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 CICS1, CICS2, and CICS3.

```
%INCLUDE CICS1  
%INCLUDE CICS2  
%INCLUDE CICS3
```

ACORESTART—NetView Restart Commands

This control file entries define commands to be executed when NetView is started or recycled with automated subsystems that are already active. They are typically used to trigger a resynchronization or initialization process. CICS Automation requires that the command CICSRSYC be called with these control file entries. (see “CICSRSYC - CICS resync” on page 113).

Comments and Usage Notes

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

Examples of Usage

Example 1

```
CICS1 ACORESTART,CMD=(,,"CICSRSYC CICS1")
```

When a startup process is initiated for subsystem CICS1, the command CICSRSYC is issued, with CICS1 passed as a parameter. This is the way it looks in the control file for SA OS/390 users. Usually this entry is created by using the SA OS/390 customization dialogs.

Automation Flags

Automation flags enable or disable automation for specific types of events. CICS Automation provides extensions to:

1. Verify service periods and trigger conditions before a startup is invoked (the EVEEEXT exit must be coded).
2. Turn automation on or off for recovery of specific events.

INITSTART	Allows a startup to be invoked for this subsystem during NetView initialization. The exit is required for this entry.
RESTART	Allows the startup process to be initiated when a startup is required other than the initial startup. The exit is required for this entry.
RECOVERY	You can specify whether or not you want automation to take place for: <ul style="list-style-type: none">• Short on storage conditions• Storage violations• Autoinstall failures• Transaction failures.

Refer to Example 3.

Comments and Usage Notes

1. Refer to the SA OS/390 customization dialogs to build these control file entries.
2. The exit EVEEEXT can still be used when AUTO=N is set. This allows CICS to be started only through the operator interface. If EVEEEXT is the last exit in the list, it issues return code '100' instead of '0'. Return code '100' indicates that the automation is off but 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 subsappl subtype .  
If autoind = 'NOAUTO' then  
    return_code = 99      /* anything but 0 */  
Else  
    return_code = 0  
Return return_code
```

3. Thresholds for special CICS recovery conditions are defined with the control file entry described in "THRESHOLDS—Error Thresholds" on page 58.

Examples of Usage

Example 1

```
AUTOMATION CICS1,AUTO=Y  
INITSTART CICS1,AUTO=E,EXITS=(EVEEEXT)  
RESTART CICS1,AUTO=E,EXITS=(EVEEEXT)
```

The AUTOMATION flag is set to Y. If you don't want automation to occur for this subsystem, change this to N. INITSTART and RESTART exits must be coded as shown.

Example 2

```

AUTOMATION CICS1,AUTO=Y,
INITSTART CICS1,AUTO=E,EXITS=(EVEETEXT),(NOAUTO=(*,00:00,24:00)
RESTART CICS1,AUTO=E,EXITS=(EVEETEXT)

```

In this example, INITSTART automation is turned off. But, because the exit is specified, you can use the operator interface to initiate an INITSTART. The exit checks service periods and trigger conditions.

Example 3

```

RECOVERY CICS1.SOS,AUTO=Y
RECOVERY CICS1.VIOL,AUTO=Y
RECOVERY CICS1.AUTO,AUTO=Y
RECOVERY CICS1.VTAMACB,AUTO=Y
RECOVERY CICS1.TRAN,AUTO=Y
RECOVERY CICS1.TRAN.HAN1,AUTO=Y

```

This example shows the five specific types of recovery that can be automated, or not automated. They are all set to YES for CICS1. Valid types are:

SOS	Short on storage. Refer to “RCVRSOS - Short-on-storage handler” on page 80 for further definitions.
VIOL	Storage violation. Refer to “RCVRVIOL - Storage violation handler” on page 82 for further definitions.
AUTO	Autoinstall failures. Refer to “RCVRAUTO - Autoinstall time limit” on page 79 for further definitions.
VTAMACB	VTAM ACB errors. No further definitions are required.
TRAN	Transaction failures (global). Refer to “RCVRTRAN - Transaction recovery” on page 81 for further definitions.
TRAN.trid	Transaction failures (specific). Refer to “RCVRTRAN - Transaction recovery” on page 81 for further definitions.

AUTOOPS—Automation Operators

The AUTOOPS entries define the NetView operator IDs that function as automation operators. These automation operators provide the ability to respond to operating system, subsystem, and network messages without requiring a human operator. Each automation operator task executes actions similar to a human operator through defined procedures and actions specified in the control file.

The automation operators are assigned specific messages to act upon. The message assignments are made in the MSG parameter of the AUTOOPS entry. The following automation operators are reserved for CICS Automation:

CICSMSTR This keyword is used to identify the automation operator that provides message distribution services for CICS subsystems. The message content is used to identify the subsystem for which this message applies. Messages are then routed to the automation operators defined with the CICSOP nn keywords, which are subsystem specific.

Enter the message prefixes when you get to the AOC/MVS Automation Operator NetView User IDs panel.

CICSOP nn These keywords are used to identify the automation operators that handle messages for specific CICS subsystems. Refer to the AUTOOPS keyword in the control file entry described in “CICSNTL - CICS extensions for SUBSYSTEM entry” on page 66 to see how the CICSOP nn keyword is associated with a specific CICS subsystem.

CICSCPPI Handles PPI communications from CICS. Refer to “CICS Automation’s Use of the Program-to-Program Interface” on page 6 for more information.

Comments and Usage Notes

1. Refer to the SA OS/390 customization dialogs to build this control file entry.
2. There are corresponding NetView DSIOPF definitions for these automation operators.
3. If security checking is to be used, automation operators need to be authorized to perform the specific tasks. Refer to “Step 3: CICS Automation Security Checking” on page 184.

Examples of Usage

Example 1

```
AUTOOPS CICSMASTER, ID=AUTCICS, MSG=(EVE*, DFH*)
AUTOOPS CICSOP01, ID=(AUTC01A)
AUTOOPS CICSOP03, ID=(AUTC03A)
AUTOOPS CICSPPPI, ID=AUTCPPI
CICSNTL CICS1, AUTOOPS=CICSOP01
CICSNTL CICS2, AUTOOPS=CICSOP01
CICSNTL CICS3, AUTOOPS=CICSOP03
```

In this example, AUTCICS is the message distribution automation operator. Messages for CICS subsystems CICS1 and CICS2 are routed to AUTC01A and messages for CICS subsystem CICS3 are routed to AUTC03A. AUTCPPI is used for PPI communications.

SDF—Status Display Facility Color Definitions

Several words are used to support the Status Display Facility CICS 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 the order in which the entries are presented.
- The color override.

The keywords are:

CICSTRAN	CICS Transactions.
CICSHLTH	Health checking.
CICSLMT	Link monitoring.
CICSTIMR	CICS timers.
CICSVIOL	Storage violations (CICS storage).
CICSSOS	Short on storage conditions (CICS storage).
VTAMACB	VTAM ACB errors.
CRITMSG	The default critical messages definition (CICS Critical Message).
CRITMSGA	Messages ending in A (CICS Critical Message).
CRITMSGE	Messages ending in E (CICS Critical Message).
CRITMSGW	Messages ending in W (CICS Critical Message).
CRITMSGI	Messages ending in I (CICS Critical Message).

Comments and Usage Notes

1. Refer to the SA OS/390 customization dialogs to build this control file entry.
2. Refer to “How to Set Up the Status Display Facility for CICS Automation” on page 14.
3. It is recommended that you use the default priority and color definitions.

Examples of Usage

Example 1

The SA OS/390 control file entries look something like this:

```
SDF    CICSTRAN,PR=200,CO=R
SDF    CICSHLTH,PR=200,CO=R
SDF    CICSLMT,PR=200,CO=R
SDF    CICSTIMR,PR=200,CO=R
SDF    CICSVIOL,PR=200,CO=R
SDF    CICSSOS,PR=200,CO=R
SDF    VTAMACB,PR=200,CO=R
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
```


ENVIRON SETUP—CICS Initialization Support

This entry is used to define the basic SA OS/390 operating characteristics. CICS Automation provides an exit (EVEEIINT) that is used to build the necessary control tables during initialization.

Examples of Usage

Example 1

```
ENVIRON SETUP, ... ),  
              EXITS=(EVEEIINT),  
              ... )
```

If this entry is omitted, CICS Automation will not initiate and the operator panels will be unable to find the CICS subsystems.

ENVIRON TIMEOUT—CICS CEMT Support

This entry is used to define the amount of time to wait for specific replies before issuing error messages. The keyword CEMT= is provided with CICS Automation to define the amount of time the CEMT interface will wait for a response. The default is 29 seconds.

Comments and Usage Notes

1. Some guidance about using the CEMT= keyword: If response time is such on your system that you are receiving CEMT timeout notifications when a true problem does not exist, use the CEMT= keyword to change the wait time.
2. Refer to the SA OS/390 customization dialogs to build this control file entry.

Examples of Usage

Example 1

```
ENVIRON TIMEOUT,CEMT=180
```

This says wait three minutes (180 seconds) for a response before issuing error messages.

Shutdown Control File Entries

Two CICS Automation commands are provided for use with the shutdown control file entries. Those commands are:

CICSSHUT This is an extended command list that determines whether or not this CICS subsystem is running with XRF so that the extensions can be called with the shutdown invocation.

The CEMT PERFORM SHUTDOWN types are passed as parameters with this command, such as NORMAL, IMMEDIATE, and TAKEOVER.

CICSSHUT is recommended for all shutdown control file entries for subsystems automated by CICS Automation. See “CICSSHUT - Shutdown processor” on page 114 for more information.

CICSPURG If you are using the control file entry described in “LISTSHUT - Transaction purging during shutdown” on page 76, then you need to code this command so that CICS Automation checks for transactions that should be purged, or specifically not purged, during a shutdown. See “CICSPURG - Purge transactions” on page 112 for more information.

CMASSHUT This is a command list that determines whether or not the CICS subsystem is running as a CICSplex SM Address Space (CMAS) so that the appropriate shutdown of the CMAS is performed. See “CMASSHUT - CICSplex SM Address Space (CMAS) Shutdown” on page 137 for more information.

Comments and Usage Notes

1. Refer to the SA OS/390 customization dialogs to build this control file entry.
2. CICSSHUT issues CEMTs across the console and may require a signon before the CICSSHUT function can be issued.

Examples of Usage

Example 1

```
CICS1 SHUTNORM,CMD=(PASS1,, 'CICSSHUT NORMAL')
```

Code your control file entry for a normal shutdown.

Example 2

```
CICS1 SHUTNORM,CMD=(PASS1,, 'CICSSHUT NORMAL CICS1'),
      CMD=(PASS2,, 'CICSPURG CICS1'),
      CMD=(PASS3,, 'MVS C CICS1')
```

For PASS1, a normal shutdown will be attempted. If not completed in the time allowed, then a CICSPURG will be performed on PASS2. If not successful, then the CICS will be cancelled on PASS3.

STARTUP—Startup Commands

This control file entry defines installation dependent commands that are executed when a startup is initiated.

Comments and Usage Notes

1. Refer to the SA OS/390 customization dialogs to build this control file entry.
2. At least one STARTUP entry must be defined for each CICS subsystem. This is a CICS Automation requirement. The SA OS/390 customization dialog includes this definition when the subsystem is defined, as described in “SUBSYSTEM—Subsystem Definition” on page 57.

Examples of Usage

Example 1

```
CICS1 STARTUP,CMD=(,,'MVS S CICSONE')
```

When a startup process is initiated for subsystem CICS1, the command MVS S CICSONE is issued.

Example 2

```
CICS1 STARTUP,CMD=(,,'MVS S JOB,N=CICSDOIT')
```

Here is just another example of a command that can be issued during the start process.

SUBSYSTEM—Subsystem Definition

This entry defines the base SA OS/390 information required for the definition of a subsystem. It does not, however, contain all of the subsystem definition keywords required for CICS Automation. The control file entry described in “CICSCNTL - CICS extensions for SUBSYSTEM entry” on page 66 provides those extra keywords.

Comments and Usage Notes

1. STARTCMD=YES is a CICS Automation requirement and must be coded in the control file as shown in the examples. Using this parameter requires that an entry is coded as described in “STARTUP—Startup Commands” on page 56. This usage note only applies to the way the entries look in the control file. It does not apply to the way that it is coded.

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

Note: The length of a CICS subsystem name is limited to 8 characters. For further information, refer to “Restrictions” on page 41.

3. IPOPTIONS=NOSTART should be specified if you want the CICS subsystem status to be set to CTLDOWN during IPL. If IPOPTIONS=START is specified, the status is changed to DOWN at IPL and a start may be performed. RECYCLOPT has a similar function when NetView is recycled.

For details, refer to *System Automation for OS/390 Customization*, section “Application Policy Object” sub-sections “Start on IPL” and “Start on Recycle”.

4. Generic application classes can be used to define attribute values for selected subsystems. This can be useful when defining many CICS subsystems having common attributes.

For details, refer to *System Automation for OS/390 Customization*, section “Creating a New Application Object”.

Examples of Usage

Example 1

```
SUBSYSTEM CICS1,JOB=CICSONE,
          PARENT=VTAM,
          STARTCMD=YES
```

This is a simple entry used to define the CICS subsystem that will be known to SA OS/390 as CICS1. The JOB= keyword is used to identify the name of the job as it is known to MVS. The PARENT= keyword defines the name of the subsystem, as it is known to SA OS/390, that must be active in MVS before this subsystem can be started.

THRESHOLDS—Error Thresholds

If the RECOVERY flag (as well as the AUTOMATION flag) is set to YES, recovery procedures are initiated as required. This is usually a process that is transparent to the operator. However, notification should be sent when repeated recovery procedures exceed expected limits, as operator intervention may be needed.

The THRESHOLD entry defines threshold levels as infrequent, frequent, or critical. 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.

Comments and Usage Notes

1. These threshold entries are coded by subsystem and must be entered directly into the %INCLUDE member. The SA OS/390 customization dialog cannot be used for these threshold entries.
2. The values entered should be such that the threshold levels are reached in a logical manner (a critical threshold should not be reached before a frequent threshold). For instance, you cannot specify '1' (as the number of errors or abends that occur in a given time period) for the critical threshold level because this implies that the first error that occurs would be already considered 'critical'.

Examples of Usage

Example 1

```
THRESHOLD CICS1.SOS,CRIT=(2,00:45),FREQ=(2,04:00),INFR=(2,24:00)
THRESHOLD CICS1.VIOL,CRIT=(2,00:45),FREQ=(2,04:00),INFR=(2,24:00)
THRESHOLD CICS1.TRAN,CRIT=(2,00:45),FREQ=(2,04:00),INFR=(2,24:00)
THRESHOLD CICS1.TRAN.HAN1,CRIT=(2,00:45),FREQ=(2,04:00),INFR=(2,24:00)
```

This example shows the four specific types of thresholds. Definitions follow:

SOS	Short on storage. Refer to “RCVRSOS - Short-on-storage handler” on page 80 for further information.
VIOL	Storage violation. Refer to “RCVRVIOL - Storage violation handler” on page 82 for further information.
AUTO	Autoinstall failures. Refer to “RCVRAUTO - Autoinstall time limit” on page 79 for further information.
TRAN	Transaction abends (global). Refer to “RCVRTRAN - Transaction recovery” on page 81 for further information.
TRAN.transid	Transaction abends (specific). Refer to “RCVRTRAN - Transaction recovery” on page 81 for further information.

CICS Automation Control File Entries

The following entries are unique to CICS Automation. These must all be coded in an %INCLUDE member using an editor.

Entry	Description
"ABCODESYSTM - System abend recovery" on page 60.	Use this entry to define actions to be taken for specific abend codes.
"ABCODETRAN - Transaction abend recovery" on page 62.	Use this entry to define actions to be taken for transaction abend codes.
"AREA - Define a set of State/Action Tables" on page 64.	Use this entry to define a set of State/Action Tables.
"CICSGROUP - Group of CICS subsystems" on page 69.	Use this control file entry to define a set of CICS subsystems that can be started or shut down from the operator interface as a group.
"CONNECTION - Link monitoring definitions" on page 70.	Use this entry to define the interregion and intersystem communication (IRC and ISC) links so that they can be monitored by CICS Automation.
"CICSCNTL - CICS extensions for SUBSYSTEM entry" on page 66.	Use this entry to define the extended subsystem information required for CICS Automation.
"EXTCOND - External conditions" on page 73.	Use this 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.
"HEALTHCHK - Health checking" on page 75.	This entry is used to define the health check routines.
"LISTSHUT - Transaction purging during shutdown" on page 76.	Use this entry to define those transactions running under this CICS subsystem that should or should not be purged during a shutdown.
"PRODUCT - Subsystem State/Action Table sets" on page 77.	Use this entry to identify a set of State/Action Tables to be used for each CICS subsystem.
"RCVRSOS - Short-on-storage handler" on page 80.	Include this entry in the control file if you want CICS Automation to take action for short on storage conditions.
"RCVRVIOL - Storage violation handler" on page 82.	Include this entry to define commands to be executed as a result of recursive storage violations.
"RCVRAUTO - Autoinstall time limit" on page 79.	Use this entry to send alerts to the operator if a CICS autoinstall takes longer than the specified time.
"RCVRTRAN - Transaction recovery" on page 81.	Use this entry to define actions to be taken when unsuccessful attempts to execute this specific transaction exceeds the defined threshold.
"SERVICE - Service period definitions" on page 83.	Use this entry to define the service periods in which this CICS subsystem is to be active.
"TRIGGER - Startup and shutdown triggers" on page 85.	Use this entry to define startup and shutdown trigger conditions.

ABCODESYSTEM - System abend recovery

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

```
ABCODESYSTEM subsys,CODE=(msg,abend1,abend2,
                        {RESTART|NORESTART})[,CODE=(msg,abend1,abend2,
                        {RESTART|NORESTART})]
.
.
.
[,CODE=(msg,abend1,abend2,{RESTART|NORESTART})]
```

Keyword and Parameter Definitions

subsys

The symbolic name by which this CICS subsystem is known to SA OS/390.

CODE=

Defines which abends are restartable, as shown in the following descriptions:

msg

The abend message ID.

abend1 and *abend2*

The specific abend codes or qualifiers.

RESTART|NORESTART

Indicates whether or not to initiate a restart for this subsystem when this specific message/abend code(s) occurs.

Comments and Usage Notes

1. Abend qualifiers vary depending upon the automation table. Refer to "CICSMSG1 (EVEMSG01)—Production NetView Initialization Automation Table" on page 247 to determine the qualifiers for each message.
2. Abend codes may be marked with an asterisk '*' to indicate that any code will be considered a match. All positions in the entry must be coded and separated by commas.
3. If a CICS message (DFHxxxxxx) is trapped and included in the ABCODESYSTEM table, then it is not usually necessary to code the corresponding IEF450I message with the same user abend code (Uxxxx) in the table. An exception to this may be DFHKE1800, which is issued so closely in time to IEF450I that it may be processed before the DFHKE1800. It is therefore recommended that you either:
 - Add both DFHKE1800 and IEF450I with U1800 to the table, or
 - Exclude DFHKE1800 from the table and from the automation table as well.

Examples of Usage

Example 1

```
ABCODESYSTEM CICS1,
  CODE=(IEF450I,S222,*,RESTART),
  CODE=(DFH0607,*,*,RESTART),
  CODE=(DFH3784,*,*,RESTART),
  CODE=(DFH0408,*,*,NORESTART)
```

In this example, a restart will be initiated for message IEF450I if the qualifier is S222. If messages DFH0607 or DFH3784 are issued, restarts will be initiated. Restarts will not be initiated for message DFH0408.

Example 1

```
ABCODESYSTEM CICS1,
  CODE=(DFHKE1800,*,*,RESTART),
  CODE=(IEF450I,*,U1800,RESTART),
  CODE=(DFH3784,*,*,RESTART),
  CODE=(DFH0408,*,*,NORESTART)
```

In this example, a restart will be initiated for message DFHKE1800, which requires that a matching entry for message IEF450I with U1800 also be coded. A restart will also be initiated for message DFH3784, but a restart will not be initiated for message DFH0408.

ABCODETRAN - Transaction abend recovery

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

```
ABCODETRAN {subsys.TRAN|subsys.TRAN.tran}  
            ,CODE=(tran,abend1,pgm,{INCLUDE|EXCLUDE})  
            [,CODE=(tran,abend1,pgm,{INCLUDE|EXCLUDE})]  
            .  
            .  
            .  
            [,CODE=(tran,abend1,pgm,{INCLUDE|EXCLUDE})]
```

Keyword and Parameter Definitions

subsys.TRAN (General case syntax)

Where *subsys* is the symbolic name by which this CICS subsystem is known to SA OS/390. Use this format when the following CODE= definitions apply to specific transaction IDs (see example 1).

subsys.TRAN.*tran* (Specific case syntax)

Where *subsys* is the symbolic name by which this CICS subsystem is known to SA OS/390 and *tran* is the transaction ID for which the following definitions apply (see example 2).

CODE=

Defines which abends are recoverable, as shown in the following descriptions:

tran

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

abend1

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

pgm

The program that abended. An asterisk (*) can be used for generic specifications.

INCLUDE|EXCLUDE

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

Comments and Usage Notes

1. Either the transaction name is specified as TRAN.*tran* or as CODE=(*tran*...). Use TRAN.*tran* when you want all of the specifications to apply to one specific transaction. Use CODE=(*tran*...) when you want to code several transactions.

Examples of Usage

Example 1

```
ABCODETRAN CICST1.TRAN,  
    CODE=(CSFE,ATNI,*,EXCLUDE),  
    CODE=(CSFE,AKC3,*,EXCLUDE),  
    CODE=(*,*,*,INCLUDE)
```

In this example, recovery will not take place for transaction CSFE if the abend code is ATNI or AKC3. Recovery will take place for all other transaction and abend codes.

Example 2

```
ABCODETRAN CICST1.TRAN.CSFE,  
    CODE=(*,ATNI,*,EXCLUDE),  
    CODE=(*,AKC3,*,EXCLUDE)
```

This is another way to specify that recovery will not take place for transaction CSFE if the abend code is ATNI or AKC3.

AREA - Define a set of State/Action Tables

Use this entry to define a set of State/Action Tables.

```
AREA {CICS|user_area_name},type=table-name,  
                                     type=table-name,  
                                     .  
                                     .  
                                     .  
                                     type=table-name
```

Keyword and Parameter Definitions

CICS|*user_area_name*

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 77.

type

A 1 to 8 character symbolic name (alpha) that represents this State/Action Table. The sample control file contains the following definitions for AREA CICS:

SOS=EVEESA01

The State/Action Table used for short on storage recovery.

VIOL=EVEESA02

The State/Action Table used for storage violation recovery.

AUTO=EVEESA03

The State/Action Table used for autoinstall problem recovery.

VTAMACB=EVEESA04

The State/Action Table used for VTAM ACB problem recovery.

Comments and Usage Notes

1. There must also be a PRODUCT entry defined. See “PRODUCT - Subsystem State/Action Table sets” on page 77.
2. Refer to “How to Use the State/Action Tables” on page 24 for more information.

Examples of Usage

Example 1

```
AREA      CICS,SOS=EVEESA01,  
          VIOL=EVEESA02,  
          AUTO=EVEESA03,  
          VTAMACB=EVEESA04
```

These are the default tables supplied with CICS Automation. Refer to the samples in Appendix C, “Sample and Source” on page 229.

Example 2

```
AREA      CICS1,SOS=TESTSA01,  
          VIOL=TESTSA02,  
          AUTO=TESTSA03,  
          VTAMACB=TESTSA04
```

In this case, a different set of tables are defined for subsystem CICS1.

CICSCNTL - CICS extensions for SUBSYSTEM entry

If you are using SA OS/390 CICS system association, CICSCNTL must be specified for all systems, whether primary or secondary.

Use this entry to define the extended subsystem information required for CICS Automation.

```
CICSCNTL subsys,APPLID={applid|(VTAM APPL name,VTAM ACBNAME)
                        |(generic-applid,specific-applid)}
                        ,MAJNODE=major-node
                        ,AUTOOPS=CICSOPnn
                        [,STRTTIMER={mm:ss|01:00}]
                        [,COLDTIMER={mm:ss|08:00}]
                        [,WARMTIMER={mm:ss|03:00}]
                        [,EMERTIMER={mm:ss|07:00}]
                        [,KEYPOINTREQ={NO|YES}]
                        [,SMFID=smf-record-number|215]
```

Keyword and Parameter Definitions

subsys

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

APPLID=

Use one of the following:

applid

If XRF=NO is coded on the SIT for this CICS subsystem, use this keyword to specify the VTAM application identifier, provided that the name of the VTAM APPL statement and the ACBNAME specified in it are the same.

(VTAM APPL name,VTAM ACBNAME)

If XRF=NO is coded on the SIT for this CICS, but the name of the VTAM APPL statement is different from the ACBNAME specified in it, code them both in this entry. The VTAM ACBNAME defined must be the same as that used by the CICS application.

(generic-applid,specific-applid)

If XRF=YES is coded on the SIT for this CICS subsystem, *generic-applid* is the name given to both the active and alternate subsystem and *specific-applid* is the specific name for this subsystem. The *specific-applid* defined must be the same as that used by the CICS application.

MAJNODE=

The VTAM node for this CICS subsystem. This is used for recovery procedures when the VTAM ACB is not active.

AUTOOPS=CICSOP*nn*

Identifies the keyword used in the control file entry described in "AUTOOPS—Automation Operators" on page 50 that defines the automation operator used for this CICS subsystem.

STRTIMER=

The time in which a startup must initiate before an alert is sent to the operator. This is expressed in minutes and seconds (*mm:ss*). The default is 1 minute.

COLDTIMER=

The time in which a cold start must complete before an alert is sent to the operator. This is expressed in minutes and seconds (*mm:ss*). The default is 8 minutes.

WARMTIMER=

The time in which a warm start must complete before an alert is sent to the operator. This is expressed in minutes and seconds (*mm:ss*). The default is 3 minutes.

EMERTIMER=

The time in which an emergency start must complete before an alert is sent to the operator. This is expressed in minutes and seconds (*mm:ss*). The default is 7 minutes.

KEYPOINTREQ=

Determines whether or not a CICS keypoint is required and whether or not Indoubt UOWs can exist. If KEYPOINTREQ=YES is specified, and a keypoint is missed, warning message EVE677 is issued during the shutdown. The status will also be changed to stopping if AOC Automation determines that an immediate restart of CICS is unlikely. Message EVE674 (keypoint not taken) or EVE687 (Indoubt UOWs) may be issued at the next CICS Automation controlled startup.

For example, when a restartable abend is detected, CICS Automation leaves the status as abending and allows normal abend restart services to restart the CICS. However, if a service period shutdown fails to take a warm keypoint, CICS Automation changes the status to stopping to alert operations of a service period shutdown failure.

SMFID=

A CICS Automation command processor creates SMF records that contain subsystem names, job names, application identifiers, event types, and event data. These records can be used for availability reporting. The SMF record number used is 215. If you don't want record number 215 used, then use this keyword to specify a different record number.

Comments and Usage Notes

1. Refer also to "SUBSYSTEM—Subsystem Definition" on page 57.

Examples of Usage

Example 1

```
CICSCNTL CICS1,APPLID=DBDCCICS,MAJNODE=VTAM1,  
          AUTOOPS=CICSOP01
```

Subsystem CICS1 is known to VTAM as DBDCCICS.

CICSCNTL - CICS extensions for SUBSYSTEM entry

Example 2

```
CICSCNTL CICS3,APPLID=(CICSGEN,CICS003),MAJNODE=VTAM1,AUTOOPS=CICSOP03  
CICSCNTL CICS4,APPLID=(CICSGEN,CICS004),MAJNODE=VTAM1,AUTOOPS=CICSOP04
```

CICS3 and CICS4 are XRF subsystems, as shown by the common generic name (CICSGEN) and the two specific names (CICS003 and CICS004).

CICSGROUP - Group of CICS subsystems

Use this control file entry to define a set of CICS subsystems that:

1. Can be started or shut down from the operator interface as a group.
2. Can be sent broadcast messages to as a group.

```
CICSGROUP groupname,MEMBER=subsys,...,MEMBER=subsys
```

Keyword and Parameter Definitions

groupname

The 1 to 8 character symbolic name which will represent this group of CICS subsystems.

MEMBER=

The symbolic name (*subsys*) by which this CICS 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. CICSGROUP statements are kept only in the domain in which they are defined; they are not forwarded to the focal point. To use a group name for a target system from the focal point, the CICSGROUP statement from the target system must also be included in the focal point control file.
3. This entry is only used for operator interface startup and shutdown and broadcast processes.
4. The requested procedure starts with the first subsystem MEMBER defined in the entry and steps through each MEMBER one at a time.
5. Service periods and external triggers 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

```
CICSGROUP CICSGRP,MEMBER=CICS1,MEMBER=CICS2,MEMBER=CICS3
```

In this example, the requested procedure starts with CICS1 and steps through the next two subsystems one at a time.

CONNECTION - Link monitoring definitions

Use this entry to define the interregion and intersystem communication (IRC and ISC) links so that they can be monitored by CICS Automation.

```
CONNECTION subsys_connid
    [,DESC=description]
    [,DAY=(dayx,nnnn-nnnn...)]
    .
    .
    .
    [,DAY=(dayx,nnnn-nnnn...)]
    [,TYPE={CICS|IMS}]
    [,ECHO=(process[,delay])]
    [,UTCDIFF=(hh:mm[, {WEST|EAST}])]
    [,INTERVAL=mm:ss]
    [,MAXREPAIR=number]
    [,RELDELAY=seconds]
    [,ACQDELAY=seconds]
    [,CRITICAL={YES|NO}]
```

Keyword and Parameter Definitions

subsys

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

connid

The 4 character symbolic name by which this link is known to this CICS subsystem. This is defined on the CICS subsystem itself (SYSID).

DESC=

The description of this link, as it is displayed on the Display Links panel. This description can be up to 16 characters long.

DAY=

Use this parameter to define the days and times that link monitoring 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 CICS 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).

CONNECTION - Link monitoring definitions

Four asterisks (****) are used to indicate that the connection stays up into the next day. This means that the previous day must end with four asterisks and the next day must begin with four asterisks.

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).

Specific days override DAILY, WEEKEND, and WEEKDAY. WEEKEND and WEEKDAY override DAILY.

TYPE=

Use this keyword when you want echoplexing to occur. TYPE= indicates the type of subsystem this link is attached to. Valid types are CICS and IMS.

ECHO=

Use this keyword when you want echoplexing to occur. The *process* parameter is used to define the transaction or programs used at the other end of the link. Refer to the following table:

If TYPE=	The process is defined as shown:
CICS	The transaction name used to call the echoplex program (EVESYCB7) on the target subsystem. The installation procedure specifies ECHO as the transaction name, but this can be changed if ECHO can not be used on the target subsystem.
IMS	Any non-blank character can be used. CICS Automation uses the IMS /TEST function to fetch a response from an IMS subsystem.

The *delay* parameter is used to define the time delay in seconds after which an echo response must have been received from the remote system before a message is issued. Valid values range from 1 to 99. The default value is 3.

UTCDIFF=

This parameter determines the time zone for which the values specified with the DAY keyword are valid. It can be used to specify the periods that monitoring is active in the remote system's time zone. Specify the time difference with the Universal Time Coordinates of the remote system's location. EAST and WEST indicates the location of the remote system's relative to the UTC (Greenwich). For example, **02:00,EAST** means two hours east of Greenwich. The default value is 0.

INTERVAL=

The time interval between 2 consecutive checks of a connection. A short interval results in an early detection of problems by the link monitor. A longer interval results in less CPU consumption by the link monitor. Valid values range from 00:05 to 99:59. The default value is 05:00.

MAXREPAIR=

Specifies the maximum number of automatic actions that are to take place after the detection of a connection failure. Valid values range from 0 to 9. The default value is 3.

RELDELAY=

Specifies the time delay in seconds needed by the system to release a connection. This can vary per connection. A connection with a system in the same domain is likely to be released faster than a connection with several

CONNECTION - Link monitoring definitions

intermediate nodes. An estimate of a proper value for this keyword can be obtained by releasing a link with the master terminal command (CEMT). Valid values range from 4 to 99. The default value is 5.

ACQDELAY=

Specifies the time delay in seconds needed by the system to acquire a connection. This can vary per connection. A connection with a system in the same domain is likely to be acquired faster than a connection with several intermediate nodes. An estimate of a proper value for this keyword can be obtained by acquiring a link with the master terminal command (CEMT). Valid values range from 4 to 99. The default value is 5.

CRITICAL=

Specifies whether or not a link failure for this link will change the severity shown on the Status Display Facility status panels.

Comments and Usage Notes

1. If echoplexing is not used, then link monitoring checks that the links are active on this CICS subsystem.
2. If DAY= is omitted, monitoring occurs whenever CICS is up and active.
3. Refer to "How to Use Link Monitoring" on page 30 for more information.

Examples of Usage

Example 1

```
CONNECTION CICS1.C101,DAY=(DAILY,0800-1800)
```

This simple example contains only the required information. The connection, as it is known to the CICS subsystem, is named C101 and it is monitored every day between 8 am to 6 pm UTC time.

Example 2

```
CONNECTION CICS1.C102,TYPE=IMS,  
          DESC='IMS1 on VALE1'  
          UTCDIFF=(05:00,WEST)  
          DAY=(WEEKDAY,0900-1100,1300-1900),  
          ECHO=(x)
```

This example, for the same subsystem (CICS1), defines information for the link known as C102. It is attached to an IMS subsystem. The description of this link, as it displays on the CICS Automation: Display Links panel, is IMS1 on VALE1. This IMS system is in a time zone that is 5 hours west of Greenwich. The connection is monitored from Monday through Friday from 9 am to 11 am and from 1 pm to 7 pm, expressed in the time zone of the IMS system. Echoplexing is performed for this connection.

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 CICS 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 CICS 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, as indicated by the following messages:

```
DFH1500 CICS/MVS VERSION 2.1 START-UP IS IN PROGRESS
DFH1500 CICS/ESA VERSION 3.2 START-UP IS IN PROGRESS
```

UP A startup is complete, as indicated by the following messages:

```
DFH1500 applid : CONTROL IS BEING GIVEN TO CICS
DFH1517 applid CONTROL IS BEING GIVEN TO CICS
```

DOWN A shutdown is complete, as indicated by the following messages:

```
IEF450I   jjj sss [ppp] - ABEND {scde|ucde} REASON=xxxxxxx TIME=hh.mm.ss
IEF404I   JOB ENDED
```

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 "CICSPPOST - Post an external event" on page 111.
2. The *event* name used here matches the *event* name described in "TRIGGER - Startup and shutdown triggers" on page 85, 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 control file entry can be used instead of the UNSET keyword to unset triggers for specific subsystems.

EXTCOND - External conditions

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 journal is offloaded'
```

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 journal is offloaded'  
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.

HEALTHCHK - Health checking

Use this entry to define the health check routines.

```
HEALTHCHK subsys,FUNCTION=(0,pgm,int,resp,{AUTO|NOAUTO},'desc')
      .
      .
      .
      [,FUNCTION=(9,pgm,int,resp,{AUTO|NOAUTO},'desc')]
```

Keyword and Parameter Definitions

0 through 9

Up to 10 health check entries can be specified for each CICS subsystem. 0 to 9 identifies a health check entry.

pgm

The program name as defined to the CICS region.

int

The interval, expressed in hours, minutes, and seconds, in which this health-check program is to be initiated. The format is *hh:mm:ss*. The maximum is 99:59:59..

Note: The interval must be greater than the response time limit.

resp

How long to wait for a response before sending an alert to the operator. This is expressed in seconds. The maximum is 120 seconds.

AUTO|NOAUTO

Code NOAUTO if you only want this health check routine to be activated through the operator interface. The default AUTO activates the routine automatically when the CICS subsystem status is changed to UP and deactivated it when the CICS subsystem terminates.

desc

The description of this health check routine as it appears on the CICS Automation: Health Checking panel. Up to 20 characters can be used.

Comments and Usage Notes

1. It is recommended that the AUTO default be used with the HEALTHCHK parameter.
2. Refer to "How to Use Health Checking" on page 28 for more information about health checking.

Examples of Usage

Example 1

```
HEALTHCHK CICS1,FUNCTION=(0,HCPAY1,00:02:00,15,, 'Check payroll database'),
      FUNCTION=(1,IMS,00:02:00,15,, 'Check IMS001')
```

There are two health check programs that will be run, HCPAY1 and IMS, and these will be run every 2 minutes. If a response doesn't come back within 15 seconds, the operator is notified.

LISTSHUT - Transaction purging during shutdown

Use this entry to define those transactions running under this CICS subsystem that should or should not be purged during a shutdown.

```
LISTSHUT subsys, {EXCLUDE|INCLUDE}=transid  
          [, {EXCLUDE|INCLUDE}=transid]  
          .  
          .  
          .  
          [, {EXCLUDE|INCLUDE}=transid]
```

Keyword and Parameter Definitions

subsys

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

EXCLUDE=

Do not purge this transaction during a shutdown.

INCLUDE=

Purge this transaction during a shutdown.

Comments and Usage Notes

1. If this control file entry is not used, no transactions are purged.
2. When the LISTSHUT control file entry is used for this subsystem, the CICS PURG command list must be coded in the shutdown control file entries for this subsystem. See "Shutdown Control File Entries" on page 55.

Examples of Usage

Example 1

```
CICS1 LISTSHUT, EXCLUDE=PAYR,  
              EXCLUDE=BAT1,  
              EXCLUDE=BAT2
```

This example specifies that PAYR, BAT1, and BAT2 are specifically not purged during a shutdown of CICS1.

Example 2

```
CICS2 LISTSHUT, INCLUDE=PAYR
```

In this example, PAYR is purged during a shutdown of CICS2.

PRODUCT - Subsystem State/Action Table sets

Use this entry to identify a set of State/Action Tables to be used for each CICS subsystem.

```
PRODUCT product, SUBSYS=(subsys, AREA={CICS|user_area_name})
      [, SUBSYS=(subsys, AREA={CICS|user_area_name})]
      .
      .
      .
      [, SUBSYS=(subsys, AREA={CICS|user_area_name})]
```

Keyword and Parameter Definitions

product

The product for which these State/Action Tables are being defined. In this case, it is CICS.

subsys

The symbolic name by which this CICS 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 64.

Comments and Usage Notes

1. There must be a **SUBSYS=** keyword defined for each CICS subsystem automated by CICS Automation.
2. The sample control file provides default definitions for AREA=CICS. If a user defined set of tables are used, the AREA control file entry must be used to define those tables.
3. Refer to “How to Use the State/Action Tables” on page 24 for more information.

Examples of Usage

Example 1

```
PRODUCT CICS, SUBSYS=(CICS1, AREA=CICS)
      SUBSYS=(CICS2, AREA=CICS)
      SUBSYS=(CICS3, AREA=CICS)
      SUBSYS=(CICS4, AREA=CICS)
```

This is how this entry would look if there were only four subsystems automated on this domain.

PRODUCT - Subsystem State/Action Table sets

Example 2

```
PRODUCT CICS,SUBSYS=(CICS1,AREA=TEST)
```

Indicates that the set of State/Action Tables known as TEST, as defined with the AREA control file entry, are used for this subsystem.

RCVRAUTO - Autoinstall time limit

Use this entry to send notifications to the operator if a CICS autoinstall takes longer than the specified time.

```
RCVRAUTO subsys.AUTO,TIMELIMIT=hh:mm:ss
```

Keyword and Parameter Definitions

subsys.AUTO

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

TIMELIMIT=

Use this keyword to determine if and when the Status Display Facility is updated with the error condition. The counter starts when the failure event occurs and stops at the completion of a successful install. If the install completes within the time specified, the Status Display Facility is not updated. If the install does not complete within the time specified, the Status Display Facility is updated.

TIMELIMIT= is expressed in hours, minutes, and seconds (*hh:mm:ss*).

Comments and Usage Notes

1. None.

Examples of Usage

Example 1

```
RCVRAUTO CICS1.AUTO,TIMELIMIT=00:05:00
```

The Status Display Facility is updated if an autoinstall stays suspended for longer than 5 minutes.

RCVRSOS - Short-on-storage handler

Use this control file entry to notify the operator when a short-on-storage condition exceeds the time limit specified with the **TIMELIMIT=** keyword.

```
RCVRSOS subsys.SOS,TIMELIMIT=mm:ss
        [,CMD=(,,'cmd')]
        .
        .
        .
        [,CMD=(,,'cmd')]
```

Keyword and Parameter Definitions

subsys.SOS

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

TIMELIMIT=

Specifies the time limit that a short-on-storage condition can exist before the commands (CMD=) are executed.

CMD=

The command or commands to be issued when the short-on-storage condition exceeds the time limit specified with the **TIMELIMIT=** keyword.

Comments and Usage Notes

1. There is a sample command list, **EVEERDMP**, that can be used when a dump is to be produced. This program is called with the **CMD=** keyword.
2. The **CRIT=** keyword in the control file entry described in “THRESHOLDS—Error Thresholds” on page 58 is used to define how many dumps are to be produced within a specific time frame before no more dumps are produced.

Examples of Usage

Example 1

```
RCVRSOS CICS1.SOS,TIMELIMIT=00:30
```

This entry states that, if the short-on-storage condition lasts longer than 30 seconds, notify the operator.

Example 2

```
RCVRSOS CICS1.SOS,TIMELIMIT=00:30,CMD=(,,'EVEERDMP')
```

This entry states that, if the short-on-storage condition lasts longer than 30 seconds, notify the operator and produce a dump.

RCVRTRAN - Transaction recovery

Use this entry to define actions to be taken when transaction abends occur.

```
RCVRTRAN {subsys.TRAN|subsys.TRAN.tran}
          ,CMD=(,,'cmd')
          [,CMD=(,,'cmd')]
          .
          .
          .
          [,CMD=(,,'cmd')]
```

Keyword and Parameter Definitions

subsys

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

tran

A specific transaction. If this is not used, then the commands apply to all transactions.

CMD=

The command or commands to be issued when the failures for this transaction exceed the threshold.

Comments and Usage Notes

1. The CRIT= keyword in the control file entry described in “THRESHOLDS—Error Thresholds” on page 58 is used to indicate how many abends can occur before automation is to take place.
2. The SA OS/390 variable EHKVAR1 is set with the name of the transaction. This allows you to tailor your commands using the transaction name, such as disabling the transaction.

Examples of Usage

Example 1

```
RCVRTRAN CICS1.TRAN,
          CMD=(,,'MSG OP1,TRAN &EHKVAR1 FAILED')
```

This entry states that the command shown is to be executed for all transactions that do not have specific entries for them. It is the default command.

Example 2

```
RCVRTRAN CICS1.TRAN.PAYR,
          CMD=(,,'MSG OP1,TRAN &EHKVAR1 FAILED PAYR')
```

This entry is specific to transaction PAYR.

RCVRVIOL - Storage violation handler

Use this control file entry to set a trace as a result of recursive storage violations or to execute other commands.

```
RCVRVIOL subsys.VIOL,CMD=([ON|OFF]),,'cmd')  
      [,CMD=([ON|OFF]),,'cmd')]  
      .  
      .  
      .  
      [,CMD=([ON|OFF]),,'cmd')]
```

Keyword and Parameter Definitions

subsys.VIOL

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

CMD=

The command or commands to be issued in the event of recursive storage violations. ON or OFF is used to set and unset a trace. Refer to the example.

Comments and Usage Notes

1. The CRIT= keyword in the control file entry described in “THRESHOLDS—Error Thresholds” on page 58 is used to indicate when to stop executing these commands based on how many errors take place in a given time frame.

Examples of Usage

Example 1

```
RCVRVIOL CICS1.VIOL,  
      CMD=(ON,, 'MVS F CICS1 CSFE DEBUG,FAQE=ON'),  
      CMD=(OFF,, 'MVS F CICS1 CSFE DEBUG,FAQE=OFF')
```

When this entry is coded this way, a trace is set on the first occurrence, then set off on the second occurrence. The third occurrence sets the trace back on and the fourth sets it off. This scenario continues until the threshold defined with the CRIT= keyword (see “THRESHOLDS—Error Thresholds” on page 58) is exceeded.

SERVICE - Service period definitions

Use this entry to define the service periods in which this CICS 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 CICS 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 CICS 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 CICS 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 (****) are used to indicate that the subsystem stays active into the next day. This means that the previous day must end with four asterisks and 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 overrides DAILY.
5. Startup and shutdown triggers can be used in conjunction with or independent of service periods. Refer to "TRIGGER - Startup and shutdown triggers" on page 85.

SERVICE - Service period definitions

Examples of Usage

Example 1

```
SERVICE CICS001,  
      DAY=(WEEKDAY,0830-1700)
```

In this example, CICS1 is up from 8:30 in the morning until 5:00 in the afternoon during the week.

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 CICS 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 can not occur unless in is within a service period window. (Refer to “SERVICE - Service period definitions” on page 83.)

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 can not 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 “CICSPOST - Post an external event” on page 111 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.

TRIGGER - Startup and shutdown triggers

Examples of Usage

Example 1

```
TRIGGER CICS1,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 CICS1,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 CICS1,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 CICS1,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.

Chapter 4. CICS Automation Routines, Commands, and Definition Members

This section is intended to help system and application programmers write programs that use CICS Automation function. The following are described:

Subroutines

These subroutines provide generic functions for expanding automation capabilities beyond those supplied and supported by CICS Automation. They are typically invoked from other programs.

Commands and common routines

These commands and common routines can be called from the control file or the message table to invoke CICS Automation routines.

Definition members

These are modified during customization to provide additional function.

Subroutines

The CICS Automation subroutines are:

“CICSBMSG - Build message processor” on page 89

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.

“CICSQRY - Name lookup” on page 91

Retrieves CICS subsystem information.

“CICSRCMD - Request a CICS function” on page 95

Provides the ability to perform commands on any CICS region, local or remote.

“CICSSEC - Invoke security checking” on page 96

Restricts NetView operators using CICS Automation to:

1. Perform functions on an installation defined set of CICS subsystems.
2. Perform an installation defined set of functions on those subsystems.

“EVESTS - Status file interface” on page 99

Allows you to retrieve and update information in the status file.

CICSBMSG - 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.

```
CICSBMSG START,domainid,taskid
CICSBMSG DATA,linetype,data
CICSBMSG END
CICSBMSG CANCEL
```

The parameters are positional.

START

This parameter is used for the first call to CICSBMSG. 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 CICSBMSG 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 CICSBMSG processor that processing is complete. It causes the processor to route previously issued data lines to the target indicated on the CICSBMSG START command. All storage obtained during processing is released.

CANCEL

This indicates to the CICSBMSG 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. CICSBMSG requires NetView-to-NetView (NNT) gateway sessions. This subroutine cannot be used in an environment where the gateway sessions are supported by RMTCMD.
2. All calls must be made from the same invocation of the same program.
3. 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 CICSMESSAGE START is received, then message **EVE698 END generated** is added to the group.

4. Single-line and multi-line messages can be mixed in the same START-END bracket.

5. The return codes are:

<i>Table 6. CICSMESSAGE Return Codes</i>	
RC	Meaning
0	OK.
4	Invalid type. Type should be START, DATA, END or CANCEL.
8	Invalid line type. Type should be START, DATA, END or CANCEL.
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.

CICSQRY - Name lookup

Use this routine to retrieve CICS subsystem information.

If a subsystem is in FALLBACK or MOVED status for a particular domain, CICSQRY with LOCAL scope does not recognize the subsystem since this routine only tracks the instance of the subsystem that is being actively managed. A CICSQRY with REMOTE or ALL scope may find the named subsystem elsewhere.

```
CICSQRY REQ={VALIDATE|DETLLIST|MSTRLIST|GET}
[,NAME={subsystem|group|domain|jobname}]
[,TYPE={CICS|GROUP|DOMAIN|ANY|JOBNAME}]
[,SCOPE={LOCAL|REMOTE|ALL}]
```

Keyword and Parameter Definitions

REQ= The request type. The request types are:

VALIDATE

A search is made for the name (NAME=) and type (TYPE=) specified so that the name can be validated. SCOPE= is only valid with this request if TYPE=CICS.

DETLLIST

A search is made for the name (NAME=) and type (TYPE=) specified so that a list of subsystems by group or domain can be built. Only TYPE=GROUP or TYPE=DOMAIN are used with DETLLIST. SCOPE= is not valid with the request.

MSTRLIST

A search is made for the type specified (TYPE=) to build a master list of subsystems of all groups or domains. Only TYPE=GROUP or TYPE=DOMAIN are used with MSTRLIST. SCOPE= is not valid with this request.

GET

CICS Automation searches for a specific CICS subsystem to retrieve the subsystem characteristics. NAME=, and SCOPE= are valid for this request.

NAME= *subsystem|group|domain|jobname*

Used with VALIDATE and DETLLIST to provide a specific group, domain or jobname. Used with GET to provide a specific subsystem value. Valid values for the NAME= variables are:

subsystem

The name by which a CICS subsystem is known to SA OS/390.

group

A group of CICS subsystem as defined to CICS Automation.

domain

The NetView domain as it is coded with the NCCFID parameter in DSIDMN.

jobname

The jobname by which a CICS subsystem is known to SA OS/390.

CICSQRY - Name lookup

TYPE=

Used to provide a specific type. The types are:

CICS (default)

Search for a specific CICS subsystem name, as it is known to SA OS/390.

ANY

Search for a CICS name first, then a domain, then a group name. If the name is longer than 5 characters the search for a domain search is bypassed. ANY is used only with VALIDATE.

DOMAIN

The NetView domain name coded in DSIDMN with the NCCFID parameter.

GROUP

The CICS subsystem group name as it is known to CICS 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=CICS. Valid scopes are:

LOCAL

Only the local system is searched for this CICS subsystem.

REMOTE

Only the focal point system is searched for this CICS subsystem.

ALL

First, the local system is searched for this CICS 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. The return codes are:

Table 7. CICSQRY Return Codes	
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:

EVELOOKUP_NAME

Unless TYPE=JOBNAME, set to the value of the NAME= parameter. If TYPE=JOBNAME, set EVELOOKUP_NAME to the subsystem name. Otherwise, set to null.

EVELOOKUP_TYPE

Set to the value of the TYPE= parameter, unless TYPE=ANY or JOBNAME, in which case it is set to CICS or DOMAIN or GROUP as appropriate.

EVELOOKUP_DCOUNT

The count of CICS subsystems found in a DOMAIN or GROUP when REQ=DETLIST

EVELOOKUP_DLIST.n

The names of the CICS subsystems found in a DOMAIN or GROUP when REQ=DETLIST

EVELOOKUP_MCOUNT

The count of DOMAINS or GROUPS found when REQ=MSTRLIST.

EVELOOKUP_MLIST.n

The names of the DOMAIN or GROUPS found when REQ=MSTRLIST.

EVELOOKUP_JOBNAME

The jobname associated with the subsystem.

EVELOOKUP_DOMAIN

The NetView domain on which SA OS/390, managing this subsystem, is running.

EVELOOKUP_AUTOOPS

The NetView automated operator that handles automation for this subsystem.

EVELOOKUP_USERVAR

The VTAM USERVAR (or generic application ID) associated with this subsystem. This is set to '*****' if a VTAM USERVAR is not defined.

EVELOOKUP_APPLID

The specific VTAM application ID associated with this subsystem.

CICSQRY - Name lookup

Refer to the following table:

Table 8. CICSQRY Input and Output Mapping									
Input			Output variables (EVELOOKUP_)						
REQ=	TYPE=	NAME= (see note 1)	NAME	TYPE	MCOUNT	MLIST.n	DCOUNT	DLIST.n (see note 4)	APPLID DOMAIN USERVAR JOBNAME AUTOOPS
VALIDATE	CICS	R	√	CICS	–	–	–	–	–
	DOMAIN	R	√	DOMAIN	–	–	–	–	–
	GROUP	R	√	GROUP	–	–	–	–	–
	ANY	R	√	–	–	–	–	–	–
MSTRLIST	DOMAIN	–	√	DOMAIN	√	√	–	–	–
	GROUP	–	√	GROUP	√	√	–	–	–
DETLIST	DOMAIN	R	√	DOMAIN	–	–	√	√	–
	GROUP	R	√	GROUP	–	–	√	√	–
GET	– (CICS implied)	R	√	CICS	–	–	–	–	√
Notes: <ol style="list-style-type: none"> 1. R indicates that the parameter is required. 2. √ indicates that the variable is set. 3. – indicates that the variable is not set. 4. DLIST.n variables are set if DCOUNT is > 0. 									

CICSRCMD - Request a CICS function

This common routine is used to perform the requested function (CMD=) on the domain where the named CICS subsystem resides, whether local or remote. The calling program does not have to be aware of where the CICS subsystem resides. It is particularly useful with single-point-of-control as CICSRCMD 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.

```
CICSRCMD NAME=name, [RESP=YES | ACK,] [OPER=operator,] CMD=cmd
```

Keyword and Parameter Definitions

NAME=

The name by which the target CICS subsystem is known to SA OS/390, as defined with the SUBSYSTEM control file entry.

RESP=

Send back a response (YES) or just send an acknowledgment (ACK).

OPER=

The operator, on the target domain, that will execute this command. If this is omitted, the automated operator defined with the CICSNTL 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

Table 9. CICSRCMD Return Codes

RC	Meaning
0	Good.
4	Subsystem name was not supplied.
8	Function to be performed was not supplied.
12	Incorrect keyword supplied.
20	Subsystem was not found on any domain.

CICSSEC - Invoke security checking

Security checking is used to restrict NetView operators using CICS Automation to:

1. Perform functions on an installation defined set of CICS subsystems.
2. Perform an installation defined set of functions on those subsystems.

Operator signon and verification are base NetView functions.

CICSSEC *NAME=subsys,FUNCTION=function*

Keyword and Parameter Definitions

NAME=

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

FUNCTION=

The function for which checking is to be performed.

Comments and Usage Notes

1. To control access to a function, a command is defined in DSICMD using the CMDMDL statement, as shown:

```
EVESX001  CMDMDL  MOD=EVESX001,RES=Y,PARSE=Y
           CMDSYN  CICSSEC
```

2. The following are defined CICS Automation functions:

INQUIRY	Can issue an INQUIRY.
STARTUP	Can issue a START.
SHUTDOWN	Can issue a SHUTDOWN.
TRIGGERS	Can use the trigger functions.
SERVICEP	Can use the service period functions.
CEMT	Can issue CEMTs.
LMT	Can use the link monitoring functions.
HEALTH	Can use the health checking functions.
SUPPORT	Can use the support functions.
EVEPICT	Can use the CICS Automation program-to-program interface functions.
BRDCAST	Can use the broadcast functions.

3. CICSSEC 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:

CICS1	KEYCLASS 1
INQUIRY	VALCLASS 1
STARTUP	VALCLASS 2
SHUTDOWN	VALCLASS 2
TRIGGERS	VALCLASS 2
SERVICEP	VALCLASS 2
CEMT	VALCLASS 2
LMT	VALCLASS 2
HEALTH	VALCLASS 2
SUPPORT	VALCLASS 2
EVEPICT	VALCLASS 2
BRDCAST	VALCLASS 2
=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.

Note: If you change any of the default KEYCLASS and VALCLASS values shipped in EVESECUR, then you must edit EVEPRFAO and add the new values to the OPCLASS statement. This will ensure that the AUTOTASKs which will be controlling various CICS regions and functions are authorized to do so. If this is not done, message EVE004I, plus EVE551E or EVE554E, may be received at STARTUP or SHUTDOWN. For example, if the AUTOTASK for a CICS is not authorized for health, then these messages will be received at shutdown:

- EVE004I SECURITY CHECK - OPID= CICSOP, KEYVALUE= HEALTH
- EVE551E UNEXPECTED RETURN CODE (RC=S) FROM EVEEMHCD IN EVEET020

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 CICS subsystems.

4. The single-point-of-control operators are security checked before attempting to perform functions on remote subsystems. Because of this, you need to define the remote subsystems in the DSICMD member located on the single-point-of-control domain.

CICSSEC - Invoke security checking

5. The return codes are:

Table 10. CICSSEC Return Codes	
RC	Meaning
0	Good.
4	Keyword not authorized.
8	Value not authorized.
12	An internal error occurred.

Examples of Usage

Example 1

```
CICSSEC NAME=CICS1,FUNCTION=INQUIRY
```

This call checks that the operator has the authority to issue an INQUIRY from the operator interface to subsystem CICS1.

Example 2

```
CICSSEC NAME=CICS1,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
```

EVESTS - Status file interface

EVESTS is the programming interface between CICS Automation and the status file. The format is as follows:

```
EVESTS ID={subsys | (subsys,SERVICE,yy/mm/dd) | (subsys,connect-id,yy/mm/dd)}
FROM={subsys | (subsys,SERVICE,yy/mm/dd) | (subsys,connect-id,yy/mm/dd)}
TO={subsys | (subsys,SERVICE,yy/mm/dd) | (subsys,connect-id,yy/mm/dd)}
```

Note: Either ID= or FROM=/TO= is mandatory. ID= and FROM=/TO= are mutually exclusive.

```
[,REQ={DISP|FLDS|DEL|REPL}]
[,RECTYPE={IND|PER|EVE|SNW}]
[,STARTDT=(mm/dd/yy, hh:mm:ss)]
[,SERVSTARTDT=(mm/dd/yy, hh:mm:ss)]
[,SERVENDDT=(mm/dd/yy, hh:mm:ss)]
[,ENDDT=(mm/dd/yy, hh:mm:ss)]
[,VER=cics-version]
[,COLDREQ={YES|NO|NONE|INIT|COLD}]
[,INDOUBTUOW={YES|NO}]
[,STARTREQ={AR|IR|IS|CS}]
[,VTAMACB={OPENED|CLOSED}]
[,XRFSTATUS={ACTIVE|STANDBY|TAKEOVER}]
[,WKPSTATUS={TAKEN|PEND|NO}]
[,RUNAPPLID=cics-applid]
[,GENAPPLID=cics-applid]
[,RUNSTARTYPE={COLD|STANDBY|WARM|EMERGENCY|LOGTERM}]
[,ABENDCODE=abend-code]
[,ABENDMSGID=abend-code]
[,EXTEVENT=(event-name, SET|RESET)]
[,SERVICEPERIOD=(hhmm-hhmm, hhmm-hhmm, ...)]
[,SYSTEMNEWS1=system-news]
[,SYSTEMNEWS2=system-news]
[,SYSTEMNEWS3=system-news]
```

Figure 10. The EVESTS Command

Keyword and Parameter Definitions

ID= Identifies the resource or resources for this command call. ID is specified as one of the following:

subsys

The CICS subsystem name as defined with the SUBSYSTEM control file entry.

subsys,SERVICE,yy/mm/dd

Used for service period records in the status file for this subsystem (*subsys*) for the date specified (*yy/mm/dd*).

subsys,connect-id,yy/mm/dd

Used for link monitoring records in the status file for this subsystem (*subsys*) for the date specified (*yy/mm/dd*).

Note: When working with service period or link monitoring overrides, the entire string of parameters following the equal sign '=' must be enclosed in parentheses '('. Also, the date format requires 2-digit numbers for the year, month and day, separated by the slash '/'. The *connect-id* specified for a link

EVESTS - Status file interface

monitoring override is the same 4-character ID which defines this connection in the CONNID portion of the CICS Automation connection control file entry.

FROM= and TO=

Specify the starting and ending points of resources. The parameters and variables used with FROM= and TO= are the same as those used for the ID= keyword, except that multiple CICS subsystems are specified.

Note: Service periods and link monitoring periods are defined in the control file. The status file only contains overrides.

REQ= Specifies one of the following request types:

DISP This returns the requested record (default). This keyword is only used with FROM=/TO=.

FLDS This returns specific field information. Fields are specified. Variable data is not used. This keyword is only used with FROM=/TO=.

REPL This replaces or adds fields within 1 or more records. The information to be replaced or added is specified, as well as the fields.

DEL This deletes the specified record or records.

RECTYPE=

Specifies the type of record on which the request is performed. The record type defaults to the keywords used in the call. Therefore, record type does not need to be specified. However, only one record type can be used for each call. The record types are:

IND All except SERVICEPERIOD=, EXTEVENT=, and SYSTEMNEWS=
PER SERVICEPERIOD=
EVE EXTEVENT=
SNW SYSTEMNEWS=

STARTDT=

Specifies the CICS subsystem start-up date and time.

SERVSTARTDT=

Specifies the CICS subsystem service start-up date and time.

SERVENDDT=

Specifies the CICS subsystem service end date and time.

ENDDT=

Specifies the CICS subsystem end date and time.

VER= Specifies the version of the CICS product executing in the particular CICS subsystem. The version can be:

21 CICS/MVS Version 2.1
32 CICS/ESA Version 3.2
33 CICS/ESA Version 3.3
41 CICS/ESA Version 4.1
51 CICS TS for OS/390 R1
52 CICS TS for OS/390 R2

COLDREQ=

Specifies whether cold start is required when the CICS subsystem is next started. This relates to the requirement of cold start in order to bring the changes of CICS resources into effect. For CICS TS this field can also reflect that an INITIAL start of CICS is required.

YES|NO

for pre CICS TS versions

NONE|INIT|COLD

for CICS TS versions

INDOUBTUOW=

Specifies whether CICS has detected that any indoubt units of work have been detected during the last shutdown of the CICS subsystem.

STARTREQ=

Specifies the STARTTYPE requested or selected for the next start-up for the given CICS subsystem. Possible values are:

AR

for AUTO start requested

IR for INITIAL start requested

IS for INITIAL start selected

CS

for COLD start selected

VTAMACB=

Specifies the status of the VTAM ACB for the particular CICS subsystem. The status can be OPENED or CLOSED.

XRFSTATUS=

Specifies the status of the CICS subsystem which belongs to a XRF pair in the context of XRF environment. The status can be:

ACTIVE

The CICS subsystem is the current active

STANDBY

The CICS subsystem is the current alternate (backup)

TAKEOVER

The CICS subsystem is in the process of taking over and will become the new active when the takeover is completed.

WKPSTATUS=

Specifies the status of whether the warm keypoint was taken during the last shutdown of the particular CICS subsystem. The status can be:

TAKEN

A successful warm keypoint was taken during the last shutdown of the CICS subsystem.

PEND

A successful warm keypoint was taken during the last shutdown of the CICS subsystem. The CICS subsystem has subsequently been started and the status is set to PENDING in preparation for the next CICS subsystem shut down.

EVESTS - Status file interface

NO A successful warm keypoint was not taken during the last shutdown of the CICS subsystem.

RUNAPPLID=

Specifies the actual VTAM applid used during the current/last execution of the particular CICS subsystem.

GENAPPLID=

Specifies the generic VTAM applid used during the current/last execution of the particular CICS subsystem.

RUNSTARTYPE=

Specifies the actual start-up type in effect for the current/last execution of the particular CICS subsystem. The start-up type can be STANDBY, COLD, WARM, EMERGENCY or LOGTERM.

ABENDCODE=

Specifies the abend code relating to the last abnormal termination of the particular CICS subsystem.

ABENDMSGID=

Specifies the abend message related to the last abnormal termination of the particular CICS subsystem.

SERVICEPERIOD=

Specifies the temporary override values of the service periods pertaining to the particular CICS subsystem on a particular date.

It can also specify the temporary overrides of the link monitoring periods pertaining to the particular CICS subsystem and the particular link to the remote subsystem.

Up to five start/stop times (*hhmm-hhmm*) can be specified.

EXTEVENT=

Specifies the name of the external event to be set or reset in the record which can store up to a maximum of 28 names. The SET action simply stores the name specified in the unused slot while the RESET action will locate the correct slot specified by the name and then replace its content with blanks. The name of external event can be up to 8 characters in length. Up to 8 occurrences of this keyword can be specified on a single EVESTS command.

SYSTEMNEWS1=, SYSTEMNEWS2=, and SYSTEMNEWS3=

Specifies up to 70 characters per entry of installation specific information to be used by the Link Monitor Tool for the particular CICS subsystem.

Blanks, commas or equal signs are not allowed in the content. It is the caller's responsibility to substitute them with other characters before invoking the EVESTS command if they do exist in the content. Otherwise it would lead to error or unpredictable result.

Comments and Usage Notes

1. The default value for the keyword not specified in the EVESTS command is blank.
2. A single EVESTS command is allowed to operate on one status record at a time. The exception is the range call.

3. Messages containing the results of the request are returned to the original requestor (operator or command list). The messages are:

EVE050I EVESTS command failed. Reason code is *rc*.

An error was detected during validity checking. The following reason codes identify the error:

- E4** Invalid keyword specified
- E5** Limit value of zero not allowed
- E6** Limit value is not numeric
- E7** Keyword without value and REPL request specified
- E8** Invalid range, FROM value is greater than TO value
- E9** Date value (in ID, FROM or TO) is not numeric
- EA** Missing separator slash in the Date value
- EB** With the range call, request is not DISP nor FLDS
- EC** ID keyword exists with the range call
- ED** FROM keyword exists but missing TO keyword
- EE** TO keyword exists but missing FROM keyword
- EF** Third value does not belong to ID, FROM or TO keyword
- F0** Second value does not belong to ID, FROM or TO keyword
- F1** Inconsistent record type (RECTYPE) implied by keywords
- F2** Length of the value exceeds the maximum allowed
- F3** Missing begin bracket for multiple values
- F4** Missing end bracket for multiple values
- F5** Missing values in multiple values. (, ,)
- F6** Premature end of values for multiple values
- F7** Maximum number of occurrences exceeded
- F8** Less than the minimum number of required parameters for EVESTS command
- F9** Conflict between the values of 2 occurrences of the same keyword; first occurrence has no value specified.
- FA** ID keyword not specified
- FB** Invalid request, other than DISP, FLDS, REPL or DEL
- FC** Invalid record type, other than IND, PER, EVE, SNW, or LMP
- FD** Conflict between explicit RECTYPE and implicit RECTYPE
- FE** Bad request, DSIPRS or DSIGET request to NetView failed.

Note: If the reason code is FE, contact your IBM Support Center (ISC).

EVE051I No record found within the above range.

There is no record with matching key qualifiers existed within the range specified by the FROM and TO keywords. It includes both the physical and logical no-record-found cases. For the values of the range, refer to EVE055I and EVE052I preceding this message. The related command is EVESTS.

EVE052I THRU ID=*to-id* RECTYPE=*type*.

This is the second message in a group of messages which display the values of all the parameters (or the specific parameters) of the specified resource id range. The *to-id* is the resource identifier which specifies the end of the range. The related command is related command is EVESTS.

EVESTS - Status file interface

EVE053I *request failed for ID=id details.*

The request for the record has failed. The *request* can be DISP (display), FLDS (display fields), REPL (replace) or DEL (delete). The *id* is the resource identifier which forms part of the key of record failed to be processed. The related command is EVESTS. The *details* provides further information about the error encountered. It can take the form of:

NO RECORD	The record requested does not exist
VSAM RC=xyyy	The VSAM RPL error codes (RPLERRCD)
ERRCODE=zz	The error detected by the logic of EVESMSTS, zz can have one of following values: <ul style="list-style-type: none">01 VSAM get error02 VSAM get update error03 VSAM put new error04 VSAM put update error05 VSAM erase error07 ID length error08 Second VSAM action not done, but request is not REPL or DEL09 Invalid request other than DISP,FLDS,REPL,DEL on initial entry0A Invalid request other than DISP,FLDS,REPL,DEL on redrive entry0B Incorrect RECTYPE specified0C Invalid operator on EXTEVENT, neither SET nor RESET0D Record is full, and no slot is empty0E VSAM get generic error

Notes:

- a. For 0D, too many events were specified to the status file. The status file must be corrected. For all others, contact your IBM Support Center (ISC).
- b. Error codes from VSAM are documented in *MVS/XA VSAM Administration: Macro Instruction Reference*.

EVE054I *request successful for ID=id.*

The request for the record has been completed successfully. The *request* can be REPL (replace) or DEL (delete). The *id* is the resource identifier which forms part of the key of the record just processed.

EVE055I *request requested for ID=id.*

This is the header for a group of messages which display the values of all the parameters (or the specific parameters) of the specified resource. The *request* can be DISP (display) or FLDS (display fields). The *id* is the resource identifier.

EVE056I ID = *id*

Part of a multi-line message group when the range is specified with the FROM and TO keywords. It indicates the resource identifier to which the information of the following EVE057I and EVE058I messages belong. The *id* is the resource identifier.

EVE057I *keyword=values*

Part of a multi-line message group starting with EVE055I. It shows the keyword with the corresponding values.

EVE058I *values*

Part of a multi-line message group starting with EVE055I. It shows the remaining part of the value if the value exceeds 200 characters in length.

EVE059I END

Identifies the end of a multi-line message group.

4. This command is not intended as a programming interface. The EVESTS should be used for display purposes only. No edit checking is done if the user manually adds or deletes service period or connection information from the status file using this command with the REQ= option. Also, manually adding service period overrides to the status file will not initiate the process of setting the timers required to make them effective. The status file is a data repository only and additional programming is required to extract the information stored there and cause CICS Automation to use it. Status file overrides should be made only from the CICS Automation operator interface panels in NetView. Incorrect use of the EVESTS command to alter or delete records from the status file may cause unpredictable results.

Examples of Usage

Example 1

To display the CICS Automation status file entries for a CICS subsystem, CICS10AA, enter:

EVESTS ID=CICS10AA

The command returns the following:

```
EVE055I DISP requested for ID=CICS10AA
EVE057I STARTDT      = 12/07/95,14:01:18
EVE057I SERVSTARTDT  = 12/07/95,14:02:29
EVE057I SERVENDDT    =
EVE057I ENDDT        =
EVE057I VER          = 41
EVE057I COLDREQ       =
EVE057I VTAMACB       = OPEN
EVE057I XRFSTATUS     = ACTIVE
EVE057I WKPSTATUS     =
EVE057I RUNAPPLID     = CICS10AA
EVE057I RUNSTARTYPE   = EMERGENCY
EVE057I ABENDCODE     =
EVE057I ABENDMSGID    =
EVE057I GENAPPLID     = CICS10A
EVE059I END
```

EVESTS - Status file interface

Example 2

To display the CICS Automation service period overrides for CICS10AA for the one day (December 8, 1995), enter:

```
EVESTS ID=(CICS10AA,SERVICE,95/12/08)
```

The command returns the following:

```
EVE055I DISP requested for ID=CICS10AA,SERVICE,95/12/08
EVE057I SERVICEPERIOD = 0745-1715,1800-2330
EVE059I END
```

Example 3

To display the CICS Automation service period overrides for CICS10AA for a seven day period (December 8 to 14, 1995), enter:

```
EVESTS FROM=(CICS10AA,SERVICE,95/12/08),  
TO=(CICS10AA,SERVICE,95/12/14)
```

The command returns the following:

```
EVE055I DISP requested for ID=CICS10AA,SERVICE,95/12/08
EVE052I THRU ID=CICS10AA,SERVICE,95/12/14 RECTYPE=PER
EVE056I ID = CICS10AA,SERVICE,95/12/08
EVE057I SERVICEPERIOD = 0745-1715,1800-2330
EVE056I ID = CICS10AA,SERVICE,95/12/09
EVE057I SERVICEPERIOD = 0745-1715,1800-2000,2030-2330
EVE056I ID = CICS10AA,SERVICE,95/12/10
EVE057I SERVICEPERIOD = 0745-1715,1800-2030
EVE056I ID = CICS10AA,SERVICE,95/12/14
EVE057I SERVICEPERIOD = 0745-1715,1800-2000
EVE059I END
```

Example 4

To display the CICS Automation record service record period override for a link with a connection ID of C06A, enter:

```
EVESTS ID=(CICS10AA,C06A,95/12/08)
```

The command returns the following:

```
EVE055I DISP requested for ID=CICS10AA,C06A ,95/12/08
EVE057I SERVICEPERIOD = 0900-1400,1500-1730,1800-2000
EVE059I END
```

Commands and Common Routines

The CICS Automation subroutines are:

“CEMTPPI - CEMT PPI short syntax” on page 109

Allows you to code a CEMT command:

1. In your own automation routines.
2. In the automation table.
3. In the control file using a control file entry that allows you to define specific commands, such as SHUTNORM, STARTUP, RCVRSOS, and so on.

“CICSDLY - Change the shutdown delay time” on page 110

This routine is used to change the default shutdown delay time specified with the SHUTDLY parameter in the SUBSYSTEM control file entry. The shutdown delay time is used during a shutdown process where user-defined shutdown commands are used and multiple passes are specified.

“CICSPOST - Post an external event” on page 111

Use this routine to set trigger conditions in the status file.

“CICSPURG - Purge transactions” on page 112

If you are using the control file entry described in “LISTSHUT - Transaction purging during shutdown” on page 76, then you need to code this command so that CICS Automation checks for transactions that should be or should not be purged during a shutdown.

“CICSRSYC - CICS resync” on page 113

The purpose of this routine is to resynchronize CICS information with what is currently operational in the system (such as the VTAM ACB status). When NetView is initialized, the routine updates the status file.

“CICSSHUT - Shutdown processor” on page 114

This is an extended command list that determines whether or not this CICS subsystem is running with XRF so that the extensions can be called with the shutdown invocation.

“EVEED003 - Critical message handler for the Status Display Facility” on page 116

This routine is used to identify those messages that are to be displayed on the Status Display Facility critical messages panels.

“EVEERDMP - CICS dump” on page 117

Creates dumps for specific CICS problems. It will dump the associated MVS region using the MVS DUMP command. It can be used for situations such as short on storage conditions and you want an MVS dump instead of a CICS internal dump.

“EVESROUT - CICS message router” on page 118

The purpose of this common routine is to distribute messages from the common CICS autotask operator, CICSMASTER, to the unique automation operators for each subsystem.

“EVEEMIGR - Migrate subsystem to CICS/TS” on page 119

Use this routine to migrate a subsystem from versions of CICS prior to CICS/TS.

“EVEEY00S - Common state handler for State/Action Tables” on page 120

This routine is used to drive actions defined in the State/Action Tables.

“CICSHLTH - Linemode Health Checking” on page 122

This command allows you to perform linemode health checking.

“CICSSTRT - Linemode CICS Startup” on page 124

This command allows you to perform linemode CICS startup.

“CICSSTOP - Linemode CICS Shutdown” on page 126

This command allows you to perform linemode CICS shutdown.

“CICSOVRD - Linemode SIT Override” on page 128

This command allows you to perform linemode SIT overrides.

“CICSLM - Linemode Link Monitor” on page 130

This command allows you to perform linemode link monitoring.

“CICSMHDR - Linemode Message Options” on page 133

This command allows you to set message header options.

“CICSTRCE - Linemode Trace Options” on page 135

This command allows you to set trace options.

“CMASSHUT - CICSplex SM Address Space (CMAS) Shutdown” on page 137

This routine invokes CICSplex SM (CPSM) Application Programming Interface calls to shut the selected CMAS down.

CEMTPPI - CEMT PPI short syntax

CEMTPPI allows you to code a CEMT command:

1. In your own automation routines.
2. In the automation table.
3. In the control file using a control file entry that allows you to define specific commands, such as SHUTNORM, STARTUP, RCVRSOS, and so on.

It accepts CEMT input as data, sends it across the program-to-program interface, and sends a response back to the originating task.

CEMTPPI *subsys* *cemt-command-stream*

Keyword and Parameter Definitions

subsys

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

cemt-command-stream

The CEMT command stream, such as SET TASK DISABLE. Do not prefix the command with CEMT.

Comments and Usage Notes

1. This command can route across domains.
2. The CEMT PERFORM SHUTDOWN option is not allowed across the program-to-program interface. Therefore, it cannot be issued with CEMTPPI.

CICSDLY - Change the shutdown delay time

This routine is used to change the default shutdown delay time specified with the SHUTDLY parameter. The shutdown delay time is used during a shutdown process where user-defined shutdown commands are used and multiple passes are specified. It indicates how long to wait for the shutdown before executing the next command in the sequence. Multiple passes are only used when the shutdown does not occur within the specified time frame.

```
subsys shuttype CMD=(passx,,CICSDLY hh:mm:ss)
```

Keyword and Parameter Definitions

subsys

The symbolic name by which this CICS subsystem is known to SA OS/390.

shuttype

The shutdown type: SHUTNORM, SHUTIMMED, and SHUTFORCE.

passx

The pass. For example, PASS1, PASS2, PASS3, and so on.

hh:mm:ss

How long to wait for the previous command request to complete before executing the command specified in the next pass.

Comments and Usage Notes

If a delay time is not specified on a particular pass, the time defaults back to the control file entry. This common routine needs to be used on each pass that you want the default delay time changed.

CICSPOST - Post an external event

Use this routine to set trigger conditions in the status file.

```
CICSPOST NAME=subsys, FUNCTION={SET|UNSET}, EVENT=event
[, TYPE={STARTUP|SHUTDOWN}]
```

Keyword and Parameter Definitions

NAME=

Is used to define the symbolic name by which this CICS subsystem is known to SA OS/390, as defined in the CICSCNTL control file entry.

FUNCTION=

Specified whether the trigger is to be SET or UNSET. If SET is specified, then STARTUP or SHUTDOWN must be specified with the TYPE=keyword.

EVENT=

The name of the external condition that represents this trigger. This name is used in the EXTCOND and TRIGGER control file entries.

TYPE=

Specifies whether this is a STARTUP or SHUTDOWN trigger. This keyword is required if FUNCTION=SET.

Comments and Usage Notes

1. The return codes are:

Table 11. CICSPOST Return Codes	
RC	Meaning
0	Function performed (normal exit)
1	Security check failed
2	Bad keyword
3	CICSQRY (lookup) failed
4	Status file update failed
5	Error on call to EVEEI111 (startup)
6	Error on call to EVEET111 (shutdown)
>10	Return code from EVEEI111 or EVEET111

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.

CICSPURG - Purge transactions

If you are using the control file entry described in “LISTSHUT - Transaction purging during shutdown” on page 76, then you need to code this command so that CICS Automation checks for transactions that should be or should not be purged during a shutdown.

CICSPURG [*subsys*]

Keyword and Parameter Definitions

subsys

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

Comments and Usage Notes

1. If a subsystem name is not specified, the TGLOBAL SUBSAPPL, which is set by AOCQRY, is used.
2. Refer to “Shutdown Control File Entries” on page 55 for more information.

Examples of Usage

Example 1

```
CICS1 SHUTDOWN,CMD=(PASS1,, 'CICSSHUT NORMAL'),  
                  CMD=(PASS2,, 'CICSPURG'),  
                  CMD=(PASS3,, 'MVS C CICS1')
```

In this example, CICSPURG is used on the second attempt to shutdown this subsystem.

CICSRSYC - CICS resync

The purpose of this routine is to resynchronize CICS information with what is currently operational in the system (such as the VTAM ACB status). When NetView is initialized, the routine will update the status file. If a CICS subsystem is active, it determines whether or not a shutdown should occur according to trigger and service period specifications. If the trigger and service period specifications indicate that this CICS should not be active, CICSRSYC will shut it down. If this CICS should be active, and health checking and link monitoring are defined for this subsystem, CICSRSYC will activate these monitoring functions. The format is:

```
CICSRSYC subsys
```

Keyword and Parameter Definitions

subsys

The symbolic name by which this CICS subsystem is known to SA OS/390.

Comments and Usage Notes

1. If *subsys* is not specified, CICSRSYC will access task global SUBSAPPL, which will probably not contain the correct value. The resync process may therefore be attempted on the wrong subsystem.
2. The preferred method to put this command in the control file is by using the AOC/MVS customization dialogs rather than by edit. The following example shows how this command will look when created by the dialogs.

Examples of Usage

Example

When is restart is initiated on CICS1, use CICSRSYC to determine the conditions under which CICS1 should or should not be active. CICS Automation will take action accordingly.

```
CICS1 ACORESTART,CMD=(,,"CICSRSYC CICS1")
```

CICSSHUT - Shutdown processor

This is an extended command list that determines whether or not this CICS subsystem is running with XRF so that the proper shutdown command can be called for the shutdown invocation.

```
CICSSHUT {NORMAL|IMMED|TAKEOVER|DUMP} [cicsname] [SDTRAN=tranid|NONE]
```

Keyword and Parameter Definitions

cicsname

The job name or subsystem name of the CICS. *cicsname* is an optional parameter.

SDTRAN=*tranid*|NONE

tranid is the name of a CICS transaction that is to run at shutdown. The specified transaction overrides the SIT SDTRAN= specification, or the default CICS supplied shutdown assist transaction CESD. If 'NONE' is specified, it will be translated into 'NOSDTRAN', meaning that no shutdown assist transaction is to run at shutdown.

The SDTRAN= parameter is optional and valid only for 'CICS TS for OS/390 V1R1' and higher versions. It is ignored for lower releases of CICS.

This routine invokes CICS transactions and the parameters perform the shutdown as described in the CICS operator manuals. If you are running in XRF and the backup system is active, CEBT is used to perform the shutdown. Otherwise, CEMT is used.

Comments and Usage Notes

1. The CEMT PERFORM SHUTDOWN types are passed as parameters to this command.
2. CICSSHUT is recommended for all shutdown control file entries for subsystems automated by CICS Automation. See "Shutdown Control File Entries" on page 55 for more information.
3. These commands are issued across the console. You may have to sign the console on first before issuing the command.

Examples of Usage

Example 1

```
CICS1 SHUTNORM,CMD=(PASS1,, 'CICSSHUT NORMAL'),  
          CMD=(PASS2,, 'CICSPURG'),  
          CMD=(PASS3,, 'MVS C CICS1')
```

A normal shutdown is being requested with the first pass.

Example 2

```
CICS1 SHUTNORM,CMD=(PASS1,, 'CICSSHUT NORMAL SDTRAN=ABCD'),  
                CMD=(PASS2,, 'CICSPURG'),  
                CMD=(PASS3,, 'MVS C CICS1')
```

A normal shutdown is being requested with the first pass, with tranid 'ABCD' to run as the shutdown transaction.

Example 3

```
CICS1 SHUTNORM,CMD=(PASS1,, 'CICSSHUT NORMAL SDTRAN=NONE'),  
                CMD=(PASS2,, 'CICSPURG'),  
                CMD=(PASS3,, 'MVS C CICS1')
```

A normal shutdown is being requested with the first pass, with no shutdown transaction to run.

EVEED003 - Critical message handler for the Status Display Facility

This routine is used to list messages in the critical message Status Display Facility panel.

EVEED003 *msgtext*

Keyword and Parameter Definitions

msgtext

The message text and message identifier passed to the critical message panel.

Comments and Usage Notes

This must be called from the automation table because it uses the jobname associated with the message.

Examples of Usage

Example 1

```
* CICS STARTUP NUCLEUS MODULE NOT FOUND - NEEDS REPLY
IF MSGID='DFH1596' & TEXT = MESSAGE
  & HDRMTYPE = '>'
  THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))
  EXEC(CMD('EVESROUT EHKEGRP AUTOTYP=START') ROUTE(ALL *))
  DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
```

Example 2

```
IF MSGID='DFH0964' & TEXT = MESSAGE
  THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))
  DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
```

EVEERDMP - CICS dump

EVEERDMP will create a dump for specific CICS problems. It will dump the associated MVS region using the MVS DUMP command. It can be used for situations such as short on storage conditions and you want an MVS dump instead of a CICS internal dump.

EVEERDMP {*jobname*|*subsys*}

Keyword and Parameter Definitions

jobname

The jobname for this CICS.

subsys

The symbolic name by which this CICS subsystem is known to SA OS/390.

Examples of Usage

Example 1

```
RCVRSOS CICS1.SOS,TIMELIMIT=00:30,CMD=(,,'EVEERDMP CICS1JOB')
```

This entry states that, if the short on storage condition lasts longer than 30 seconds, notify the operator and produce a dump.

EVESROUT - CICS message router

The purpose of this common routine is to distribute messages from the common CICS autotask operator, CICSMSTR, to the unique automation operators for each subsystem.

```
EVESROUT [JOBNAME=jobname]clist [clist-parms]
```

Keyword and Parameter Definitions

JOBNAME=

The name of this MVS job. This is used for IEF403I and IEF404I messages when a jobname is specified in the message text.

clist

The command list to execute.

clist-parms

Any parameters required by the command list.

Comments and Usage Notes

1. EVESROUT can only be invoked from the NetView Automation Table.

Examples of Usage

Example 1

The following is used to trap the CICS version number:

```
* TRAP CICS VERSION NUMBER
IF MSGID='DFH1500'
  & TOKEN(4)='VERSION' & TOKEN(5)=VERNUM & TOKEN(6) = 'START-UP'
THEN EXEC(CMD('EVESROUT EVEEI004 'VERNUM) ROUTE(ALL *))
  DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
```

Example 2

The following is used to determine the startup type:

```
* CICS STARTUP TYPE (COLD/WARM/EMER,ETC.)
IF (MSGID='DFH1500' | MSGID='DFH1502' ) &
  TOKEN(3)='CICS' & TOKEN(4) = 'START-UP' &
  TOKEN(5)='IS' & TOKEN(6) = STARTTYP
THEN EXEC(CMD('EVESROUT EVEEI006 'STARTTYP) ROUTE(ALL *))
  DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
```

EVEEMIGR - Migrate subsystem to CICS/TS

Use this routine to migrate a subsystem from versions of CICS prior to CICS/TS.

```
EVEEMIGR NAME=subsys,NEWVERSION=version
```

Keyword and Parameter Definitions

NAME=

The name by which the CICS subsystem is known to SA OS/390, as defined/coded in the ACF.

NEWVERSION=

Is used to define the new version of CICS/TS that is to be migrated to. Valid versions are:

- V1R1
- V1R2

Comments and Usage Notes

1. The return codes are:

Table 12. EVEEMIGR Return Codes

RC	Meaning
0	Function performed (normal exit)
1	Invalid keyword entered
2	Invalid CICS system NAME entered
3	Invalid NEWVERSION entered
4	The specified CICS system name does not exist
>4	An unexpected error occurred and processing has not completed.

2. This command is only applicable when migrating to CICS/TS V1R1 and V1R2.
3. The version entries in both the status file and the currently active Cglobal for the selected system will be modified.

Examples of Usage

Example

The following command must be issued after migrating CICS/TS from Version 4 to CICS/TS V1R1.

```
EVEEMIGR NAME=CICSA,NEWVERSION=V1R1
```

EVEEY00S - Common state handler for State/Action Tables

This routine is used to drive actions defined in the State/Action Tables.

```
EVEEY00S [MSGID=msgid]  
          [,JOB=jobname]  
          [,MSGSTR=msgstring]
```

Keyword and Parameter Definitions

MSGID=

The message passed to the State/Action Table as an event.

JOB=

The jobname associated with the event.

MSGSTR=

The message string associated with the message. MSGSTR= cannot be coded unless MSGID= is coded.

Comments and Usage Notes

1. If MSGID= is not specified, the message that invoked the routine is used as the event.
2. If JOB= is not specified, the jobname associated with the message is used.
3. If the routine is **not** invoked from the automation table, MSGID= and JOB= **must** be coded.
4. Refer to "How to Use the State/Action Tables" on page 24.
5. Refer also to the examples in:
 - "EVEESA01—Short-on-Storage State/Action Table" on page 284.
 - "EVEESA02—Storage Violation State/Action Table" on page 286.
 - "EVEESA03—Autoinstall Problem Recovery State/Action Table" on page 287.
 - "EVEESA04—VTAM ACB Problem Recovery State/Action Table" on page 288.

Examples of Usage

Example 1

```
* VTAMACB STATE ACTION - VTAM DISCOVERED DOWN BY CICS
IF (MSGID = 'DFH3463I'
   | MSGID = 'DFHSI1572'
   | MSGID = 'DFH1572' )
& (TEXT = . 'X''52'' ' . | TEXT= . 'X''5C'' ' .
   | TEXT = . '=52.' . | TEXT= . '=5C.' .)
THEN EXEC(CMD('EVESROUT EVEEY00S MSGID=VTAMDN') ROUTE(ALL *));
```

Example 2

```
* VTAMACB STATE ACTION - ACB CLOSED BY CICS
IF MSGID = 'DFH2316I'
THEN EXEC(CMD('EVESROUT EVEEY00S MSGID=ACBCLS') ROUTE(ALL *));
```

CICSHLTH - Linemode Health Checking

CICSHLTH allows an operator or user-written routine to control health checking without using the SA OS/390-CICS panels.

```
CICSHLTH NAME=subsys, ACTION=START, PROGRAM=programe

        ACTION=STATUS, PROGRAM=programe

        ACTION=RESUME, PROGRAM=programe

        ACTION=SUSPEND, PROGRAM=programe

        ACTION=STOP, PROGRAM=programe

        ACTION=CHECK, PROGRAM=programe
```

Keyword and Parameter Definitions

subsys

The name of the CICS subsystem.

programe

The name of the user-written health check program.

ACTION=START

Used to initiate health check processing.

ACTION=STATUS

Used to determine if the health check program is active or inactive, and normal or abnormal regarding its most recent execution.

ACTION=RESUME

Used to continue a process that was temporarily suspended.

ACTION=SUSPEND

Used to temporarily stop a process.

ACTION=STOP

Used to stop health check processing.

ACTION=CHECK

Used to submit a status check regardless of the status and scheduled time interval of the health check program.

Comments and Usage Notes

Actions can only be performed using programs that are predefined in the Automation Control File.

Examples of Usage

Example 1 - Status on a Health Check Program

Command:

```
CICSHLTH NAME=CICS01A,ACTION=STATUS,PROGRAM=EVECHLTH
```

Message Response:

```
C A0F01    EVE441I  HEALTH CHECK PROGRAM EVECHLTH STATUS INACTIVE
C A0F01    EVE445I  ABNORMAL RESPONSE ON 05/25/95 09:51:35 -
```

Example 2 - Start of a Health Check Program

Command

```
CICSHLTH NAME=CICS01A,ACTION=START,PROGRAM=EVECHLTH
```

Message Response:

```
C A0F01    EVE436I  CICS01A HEALTH START FOR EVECHLTH SUCCESSFUL.
```

CICSSTRT - Linemode CICS Startup

CICSSTRT allows you to invoke the SA OS/390-CICS startup routine while bypassing certain error codes.

```
CICSSTRT NAME=subsys,STARTTYPE=type,  
          [BYPASS=(ALL,SERVICE,TRIGGERS,WARMKEYPT,STATUS  
                  PARENTCHK,INDOUBTUOW,INITIAL,*nn)]  
          [OVERRIDE=%data%]
```

Keyword and Parameter Definitions

subsys

The name of the CICS subsystem.

type

The type of startup. Startup can be default, auto, cold, emergency, standby, logterm, or initial. The types include:

- **DEFAULT**—Uses the start option specified in the SIT or the JCL.
- **AUTO**—Uses Restart Data Set to determine start up type.
- **COLD**—A cold start is done.
- **EMERGENCY**—An emergency start is done. Not applicable to CICS Version 4 or higher.
- **STANDBY**—A start is done for an XRF backup.
- **LOGTERM**—The system is started and immediately shut down. Not applicable to CICS TS.
- **INITIAL**—Similar to a CICS COLD start with additional processing for CICS TS resources. Applicable to CICS TS only.

BYPASS=(ALL,SERVICE,TRIGGERS,WARMKEYPT,STATUS,PARENTCHK,INDOUBTUOW,INITIAL,*nn)

BYPASS is used to ignore certain conditions which would otherwise inhibit the startup. The conditions ignored include:

- **ALL**—All exceptional conditions below except the subsystem is already running.
- **SERVICE**—Service periods in effect for the subsystem.
- **TRIGGERS**—Triggers set or unset for the subsystem.
- **WARMKEYPT**—Warm keypoint taken during previous execution of the CICS subsystem.
- **STATUS**—Current status of subsystem, excluding UP or EXTSTART (overrides status check).
- **PARENTCHK**—Current status of subsystem's parent (overrides parent status check).
- **INDOUBTUOW**—Indoubt units of work detected during the last shut down of CICS.
- **INITIAL**—An INITIAL start for CICS has been requested.

- **nn*—Any specific return code from startup processing (except codes that cannot be overridden).

data

Data is any keyword=value override that is supported by CICS/ESA. (It must be delimited.) An examples include PLTPI=01 where 01 is the suffix of the Program List Table that is run Post Initialization.

Comments and Usage Notes

CICSSTRT issues the STARTUP command that is found in the Automation Control file for the named CICS subsystem.

Examples of Usage

Example—Start of a CICS Subsystem

Command:

```
CICSSTRT NAME=CICS01E,STARTTYPE=DEFAULT,BYPASS=ALL
```

Message Response:

```
U A0F01      A0F570I 13:46:07 : ISSUED "MVS S CICS401E" FOR SUBSYSTEM
              CICS01E - MSGTYP IS STARTUP
```

CICSSTOP - Linemode CICS Shutdown

CICSSTOP allows you to invoke the SA OS/390-CICS shutdown routine while bypassing certain error codes.

```
CICSSTOP NAME=subsys,SHUTTYPE=type,  
           [RESTART=[NO|YES|CTL]]  
           [BYPASS=(ALL,SERVICE,TRIGGERS,  
                   STATUS,*nn)]
```

Keyword and Parameter Definitions

subsys

The name of the CICS subsystem.

type

The type of shutdown (normal, immed, or force).

RESTART=[NO|YES|CTL]

Used to determine if the CICS is to be restarted or not.

NO

This is the default and the CICS subsystem on completion of the shutdown will be placed in an AUTODOWN status.

YES

On completion of the shutdown the CICS subsystem will be placed in a RESTART status and restarted.

CTL

On completion of the shutdown the CICS subsystem will be placed in a CTLDOWN status.

CTL is used when service periods and triggers are the mechanism that is expected to restart the CICS subsystem at a later point in time.

BYPASS=(ALL,SERVICE,TRIGGERS,STATUS,*nn)

BYPASS is used to ignore conditions that may arise during the shutdown process. For example, if a CICS is within its service period and a line-mode shutdown is desired, BYPASS=SERVICE, will ensure that the shutdown takes place. The conditions ignored include:

- ALL—All exceptional conditions below except the subsystem is already running.
- SERVICE—Service periods in effect for the subsystem.
- TRIGGERS—Triggers set or unset for the subsystem.
- STATUS—Current status of subsystem (overrides status check).
- *nn—Any specific return code from stop processing (except codes that cannot be overridden).

Comments and Usage Notes

CICSSTOP invokes SHUTSYS, which in turn invokes the SHUTNORM, SHUTIMMED, or SHUTFORCE command that is found in the Automation Control file for the named CICS subsystem.

Examples of Usage

Example—Stop of a CICS Subsystem within its service period

Command:

```
CICSSTOP NAME=CICS01A,SHUTTYPE=IMMED,RESTART=NO
```

Message Response:

```
C A0F01    EVE453I  CICS01A subsystem is within its service period
```

Example—Stop of a CICS Subsystem with override

Command:

```
CICSSTOP NAME=CICS01A,SHUTTYPE=IMMED,RESTART=NO,  
          BYPASS=(SERVICE,TRIGGERS)
```

Message Response:

```
U A0F01    A0F254I  A0FS0002 SHUTDOWN NOW IN PROGRESS  
C A0F01    EVE310I  OPERATOR SHUTDOWN OF CICS01A BY GULLA IN  
              DOMAIN A0F01
```

CICSOVRD - Linemode SIT Override

CICSOVRD allows you to set CICS SIT override conditions prior to CICS startup. (Otherwise, AOC/CICS would only allow you to set override conditions through the operator startup panel or through the CICSSTRT command.)

```
CICSOVRD NAME=subsys,STARTTYPE=type,  
          ACTION=SET,OVERRIDE=%data%,  
          KEYPOINT=[REQuired|OPTional]  
  
          ACTION=STATUS
```

Keyword and Parameter Definitions

type

The type of startup.

subsys

The name of the CICS subsystem.

ACTION=SET

Used to change the SIT override for a CICS subsystem.

ACTION=STATUS

Used to inquire about the SIT options for a CICS subsystem.

data

The override data to be used to override the SIT options. (It must be delimited.)

KEYPOINT=[REQuired|OPTional]

Used to specify if a warm keypoint is required for the CICS subsystem, before these overrides can be used.

Comments and Usage Notes

The CICSOVRD command does not start the named CICS. The overrides are saved and used for subsequent starts of the CICS. The SIT overrides can be displayed by using option 2 from the CICS Automation operator panels or by using the **ACTION=STATUS** option.

This linemode command returns the following message to the invoking routine:

```
C A0F01    EVE556I  CICSOVRD Completed successfully
```

To clear an override, enter:

```
CICSOVRD NAME=subsys,ACTION=SET,OVERRIDE=%%
```

Examples of Usage

Example—Add an override for the PLTPI for CICS01A

Command:

```
CICSOVRD NAME=CICS01A,ACTION=SET,OVERRIDE=%PLTPI=02%
```

Message Response:

```
C A0F01    EVE556I  CICSOVRD Completed successfully
```

Example—Change the Keypoint option for CICS01A

Command:

```
CICSOVRD NAME=CICS01A,ACTION=SET,KEYPOINT=OPT
```

Message Response:

```
C A0F01    EVE556I  CICSOVRD Completed successfully
```

CICSLM - Linemode Link Monitor

CICSLM provides a linemode interface for the link monitoring functions of CICS Automation.

```
CICSLM NAME=subsys,ACTION=STARTLMT

        ACTION=STATUSLMT

        ACTION=CONNINFO

        ACTION=STOPLMT

        ACTION=STATUS,CONNECTION=conname

        ACTION=RECOVER,CONNECTION=conname

        ACTION=SUSPEND,CONNECTION=conname,
                                [FUNCTION=[ECHO|MONITOR]]
        ON=RESUME,CONNECTION=conname,
                                [FUNCTION=[ECHO|MONITOR]]
```

Keyword and Parameter Definitions

subsys

The name of the CICS subsystem.

ACTION=STARTLMT

Used to start link monitoring processing.

ACTION=STATUSLMT

Used to determine if link monitoring is started or stopped.

ACTION=CONNINFO

Used to get information on connection information.

ACTION=STOPLMT

Used to stop link monitoring processing.

ACTION=STATUS

Used to get status on a named connection.

ACTION=RECOVER

Used to start a series of repair actions for a named connection.

ACTION=SUSPEND

Used to temporarily stop link monitoring processing.

ACTION=RESUME

Used to restart link monitoring processing after it was suspended.

conname

This is the 4 character CICS name for a connection

FUNCTION=ECHO

Used to run the echoplexing transaction over the named connection.

FUNCTION=MONITOR

Used to run monitoring for the named connection.

Comments and Usage Notes

ACTION=STATUS, RECOVER, SUSPEND, and RESUME require the use of the CONNECTION keyword.

Examples of Usage**Example—Get connection information for CICS01A**

Command:

```
CICSLM NAME=CICS01A,ACTION=CONNINFO
```

Message Response:

```
C A0F01    EVE793I  AUTOMATION DISPLAY - CONNINFO
C A0F01    EVE794I  CURRENT ITEM - CONNID=C10A
C A0F01    EVE795I  DATA IS APPLID=CICS10AA
C A0F01    EVE795I  DATA IS DESCRIPTION=FROM CICS01A TO
C A0F01    EVE795I  DATA IS DESIRED=DOWN
C A0F01    EVE795I  DATA IS ACTUAL=UNKNOWN
C A0F01    EVE795I  DATA IS MONITOR=ON
C A0F01    EVE795I  DATA IS LASTCHK=
C A0F01    EVE795I  DATA IS ECHOPLEX=
C A0F01    EVE796I  END OF CONNINFO DISPLAY
```

Example—Get status information for specific connection

Command:

```
CICSLM NAME=CICS01A,ACTION=STATUS,CONNECTION=C10A
```

Message Response:

```
C A0F01  EVE793I  AUTOMATION DISPLAY - STATUS
C A0F01  EVE794I  CURRENT ITEM - CONN=C10A
C A0F01  EVE795I  DATA IS LOCAL=CICS01A
C A0F01  EVE795I  DATA IS REMOTE=CICS10AA
C A0F01  EVE795I  DATA IS DESCRIPTION=FROM CICS01A TO
C A0F01  EVE795I  DATA IS CONNTYPE=LU62
C A0F01  EVE795I  DATA IS CRITICAL=NO
C A0F01  EVE795I  DATA IS TIMEZONE=00:00 EAST
C A0F01  EVE795I  DATA IS MONSTATUS=ON
C A0F01  EVE795I  DATA IS ECHOSTATUS=
C A0F01  EVE795I  DATA IS LASTCHK=
C A0F01  EVE795I  DATA IS RESPONSE=
C A0F01  EVE795I  DATA IS DESTLINK=DOWN
C A0F01  EVE795I  DATA IS ACTIVELINK=UNKNOWN
C A0F01  EVE795I  DATA IS SERVICE=UNKNOWN
C A0F01  EVE795I  DATA IS ACQUIRE=UNKNOWN
C A0F01  EVE795I  DATA IS INTERVAL=27:00
C A0F01  EVE795I  DATA IS REPAIR=3
C A0F01  EVE795I  DATA IS RD=05
C A0F01  EVE795I  DATA IS AD=05
C A0F01  EVE795I  DATA IS ED=
C A0F01  EVE795I  DATA IS SYSTEM=CICS
C A0F01  EVE795I  DATA IS ECHOPROC=
C A0F01  EVE796I  END OF STATUS DISPLAY
```

Example—Resume monitoring for a specific connection

Command:

```
CICSLM NAME=CICS01A,ACTION=RESUME,CONNECTION=C10A,FUNCTION=MONITOR
```

Message Response:

```
C A0F01  EVE968I  LINK MONITORING REQUEST RESUME(MONITOR) FOR
CONNECTION C10A WAS SUCCESSFUL
```


CICSMHDR - Linemode Message Options

CICSMHDR allows you to change the message header options which display on the operator panel.

```
CICSMHDR NAME=[domain],ACTION=SET
                        OPID=opname
                        DOMAIN=[ON|OFF]
                        OPERATOR=[ON|OFF|DEFAULT]

                        ACTION=STATUS
                        OPID=opname
```

Keyword and Parameter Definitions

[*domain*]

The name of the NetView domain.

ACTION=SET

Used to change the message options for a domain or operator.

ACTION=STATUS

Used to inquire about the message options for a domain or operator.

opname

The NetView operator ID.

DOMAIN=[ON|OFF]

For ACTION=SET, this keyword turns message ID on or off for the domain and/or operator. Both can be set in a single command.

OPERATOR=[ON|OFF|DEFAULT]

For ACTION=SET, this keyword turns message ID on or off for the operator specified on the OPID keyword. A value of default allows the message id options to be controlled by the domain setting.

Comments and Usage Notes

If the NAME keyword is not supplied a default of the current domain is used.

For ACTION=SET, both the OPID and OPERATOR keywords must be used if either is specified.

For ACTION=STATUS, no additional keywords are required.

This linemode command returns the following message to the invoking routine:

```
C A0F01    EVE449I  CICSMHRD NAME=A0F01 DOMAIN=OFF OPID=  OPERATOR=
```

Examples of Usage

Example—Turn message ID on for an operator

Command:

```
CICSMHDR ACTION=SET,OPID=GULLA,OPERATOR=ON
```

Message Response:

```
C A0F01    EVE449I  CICSMHDR NAME=A0F01 DOMAIN=OFF OPID=GULLA
                OPERATOR=ON
```

Example—Turn message id on for a domain

Command:

```
CICSMHDR ACTION=SET,DOMAIN=ON
```

Message Response:

```
C A0F01    EVE449I  CICSMHDR NAME=A0F01 DOMAIN=ON OPID=  OPERATOR=
```

Example—Reset message options for a domain and operator

Command:

```
CICSMHDR ACTION=SET,DOMAIN=OFF,OPID=GULLA,OPERATOR=DEFAULT
```

Message Response:

```
C A0F01    EVE449I  CICSMHDR NAME=A0F01 DOMAIN=OFF OPID=GULLA
                OPERATOR=DEFAULT
```

CICSTRCE - Linemode Trace Options

CICSTRCE allows you to change the trace options for SA OS/390-CICS.

```
CICSTRCE NAME=[domain],ACTION=SET
                        OPID=opname
                        DOMAIN=[ON|OFF]
                        OPERATOR=[ON|OFF|DEFAULT]

                        ACTION=STATUS
                        OPID=opname
```

Keyword and Parameter Definitions

[*domain*]

The name of the NetView domain.

ACTION=SET

Used to change the trace options for a domain or operator.

ACTION=STATUS

Used to inquire about the trace options for a domain or operator.

opname

The NetView operator ID.

DOMAIN=[ON|OFF]

For ACTION=SET, this keyword turns trace on or off for the domain and/or operator. Both can be set in a single command.

OPERATOR=[ON|OFF|DEFAULT]

For ACTION=SET, this keyword turns trace on or off for the operator specified on the OPID keyword. A value of default allows the trace options to be controlled by the domain setting.

Comments and Usage Notes

If the NAME keyword is not supplied a default of the current domain is used.

For ACTION=SET, OPID and OPERATOR keywords must be used if one is specified.

For ACTION=STATUS, no additional keywords are required.

This linemode command returns the following message to the invoking routine:

```
C A0F01    EVE449I  CICSTRCE NAME=A0F01 DOMAIN=OFF OPID=  OPERATOR=
```

Examples of Usage

Example—Start trace for an operator

Command:

```
CICSTRCE ACTION=SET,OPID=GULLA,OPERATOR=ON
```

Message Response:

```
C AOF01    EVE000I 05/26/95 10:53:03 EVEEJ201 EXIT  RC= 0
C AOF01    EVE000I 05/26/95 10:53:03 EVEMSGF ENTRY 449
C AOF01    EVE000I 05/26/95 10:53:03 EVEMSG1 ENTRY
          449,,ON,,CICSTRCE,AOF01,OFF,GULLA,ON
C AOF01    EVE000I 05/26/95 10:53:04 EVEMSG1 EXIT  RC= 0
C AOF01    EVE000I 05/26/95 10:53:04 EVEMSGF EXIT  RC= 0
C AOF01    EVE449I  CICSTRCE NAME=AOF01 DOMAIN=OFF OPID=GULLA
          OPERATOR=ON
C AOF01    EVE000I 05/26/95 10:53:04 EVEEJ202 EXIT
```

Example—Start trace for a domain

Command:

```
CICSTRCE NAME=AOF01,ACTION=SET,DOMAIN=ON
```

Message Response:

```
C AOF01    EVE449I  CICSTRCE NAME=AOF01 DOMAIN=ON OPID=  OPERATOR=
```

Example—Reset trace for a domain and operator

Command:

```
CICSTRCE NAME=AOF01,ACTION=SET,DOMAIN=OFF,OPERATOR=DEFAULT,
          OPID=GULLA
```

Message Response:

```
C AOF01    EVE449I  CICSTRCE NAME=AOF01 DOMAIN=OFF OPID=GULLA
          OPERATOR=DEFAULT
```

CMASSHUT - CICSplex SM Address Space (CMAS) Shutdown

This is a command list that determines whether or not this CICS subsystem is running a CICSplex SM Address Space (CMAS). It then uses the CPSM REXX API to shut down the CMAS.

```
CMASSHUT [masname]
```

Keyword and Parameter Definitions

masname

The job name or subsystem name of the CICSplex SM Address Space (CMAS). *masname* is an optional parameter. If *masname* is not specified, the SUBSAPPL task global value is used.

This routine invokes CICSplex SM (CPSM) Application Programming Interface calls to shut the selected CMAS down.

Comments and Usage Notes

CMASSHUT is intended as a shutdown command to be defined as a shutdown pass in the automation control file. It is recommended that this be used to shutdown CICSplex SM Address Space (CMAS) subsystems. See “Shutdown Control File Entries” on page 55 for related information.

Examples of Usage

Example

```
CMAS1 SHUTNORM,CMD=(PASS1,, 'CMASSHUT'),  
      CMD=(PASS2,, 'MVS C CMAS1')
```

A normal shutdown of the CMAS is being requested with the first pass.

Definition Members

The definition members are:

“EVESPINM - CICS PPI initialization member” on page 139.

This member is used on the CICS side of the program-to-program interface 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 the program-to-program interface receiver identifier for the EVESNPPI NetView subtask program.
3. Provide the identifier of the console on which the long-running COPC transaction is started.
4. Define the relationships between function names and CICS transaction names.

“EVENTASK - NetView PPI initialization member” on page 141

This member is used on the NetView side of the program-to-program interface 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 request programs and autotasks to be used for specific functions.
3. Define response programs and autotasks to be used for specific functions.
4. Define the program-to-program interface receiver identifier for the EVESNPPI NetView subtask program.

“EVESCMT2 - Message exit table for CICS Version 2” on page 143

Identifies which transient data queue messages require automation for CICS Version 2.

“EVESCMT3 - Message exit table for CICS Version 3 and higher” on page 145.

Identifies which transient data queue messages require automation for CICS Version 3 and higher.

EVESPINM - CICS PPI initialization member

This member is used on the CICS side of the program-to-program interface 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 the program-to-program interface receiver identifier for the EVESNPPI NetView subtask program.
3. Provide the identifier of the console on which the long-running COPC transaction is started.
4. Define the relationships between function names and CICS transaction names.

Note: There is a corresponding initialization member on the NetView side. See “EVENTASK - NetView PPI initialization member” on page 141.

Following is an example of the information contained in the EVENTASK program-to-program interface initialization member:

EVEMPINM TYPE=INITIAL,	INITIAL ENTRY
BUFFQL=4,	BUFFER QUEUE LIMIT
RECEIVERID=NETVCPPI,	NPDS RECEIVER IDENTIFICATION
USERID=[YES NO],	INVOKE TRANSACTION WITH REAL USERID
CONSOLE=	TERMID OF CONSOLE FOR COPC
EVEMPINM TYPE=ENTRY,	DEFINE A FUNCTION
FUNCTION=CEMT,	FUNCTION NAME
TRANSID=COMT	TRANSACTION NAME
EVEMPINM TYPE=ENTRY,	DEFINE A FUNCTION
FUNCTION=LMT,	FUNCTION NAME
TRANSID=COLR	TRANSACTION NAME
EVEMPINM TYPE=ENTRY,	DEFINE A FUNCTION
FUNCTION=HEALTH,	FUNCTION NAME
TRANSID=COHR	TRANSACTION NAME
EVEMPINM TYPE=ENTRY,	DEFINE A FUNCTION
FUNCTION=TEST,	FUNCTION NAME
TRANSID=TSTL	TRANSACTION NAME
EVEMPINM TYPE=FINAL	REQUIRED END

Keyword and Parameter Definitions

TYPE=

Indicates the type of entry this is. Valid types are:

INITIAL

The first EVEMPINM type specified. Only one INITIAL entry can be specified.

ENTRY

This type associated a function with a CICS transaction.

FINAL

Indicates that this is the final entry. Only one FINAL entry can be specified.

BUFFQL=

Specifies the buffer queue limit for the CICS receiver side of the program-to-program 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. This keyword is only valid with TYPE=INITIAL.

RECEIVERID=

Specifies the identifier of the NetView receiver. If this keyword is omitted, NETVCPPI is assumed. This keyword is only valid with TYPE=INITIAL.

USERID=[YES|NO]

Specifies that the transaction will be invoked with the NetView user ID that invoked the PPI process. The default is NO.

CONSOLE=

Specifies the 1 to 4-character terminal identifier of the console on which the long-running COPC transaction is started. If this specification is omitted, COPC is started without a terminal. This keyword is only valid with TYPE=INITIAL.

FUNCTION=

The name of the function to be executed. The function name can be from 1 to 8 characters and must not start with the characters EVE.

TRANSID=

The name of the CICS transaction associated with this function. This transaction will be executed when the function is requested.

Comments and Usage Notes

1. A function name may not start with EVE.
2. EVESPINM must be link-edited in to one of the CICS DFHRPL libraries.
3. At least one valid TYPE=ENTRY must be specified.
4. There must be a TYPE=ENTRY definition for each function that uses the CICS Automation program-to-program interface. The corresponding NetView side initialization member entry looks like this:

```
SERVER=RESPONSE,CEMT,AUTCPPI,EVESNRSP
```

where CEMT is the function.
5. If you are running CICS Automation in more than one NetView domain on the same MVS system, then you need to provide unique RECEIVERIDs in this member. This must also be changed in the corresponding NetView program-to-program interface initialization member. See "EVENTASK - NetView PPI initialization member" on page 141, which explains where the matching RECEIVERID is changed for that member.

EVENTASK - NetView PPI initialization member

This member is used on the NetView side of the program-to-program interface 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 request programs and autotasks to be used for specific functions.
3. Define response programs and autotasks to be used for specific functions.
4. Define the program-to-program interface receiver identifier for the EVESNPPI NetView subtask program.

Note: There is a corresponding initialization member on the CICS side. See "EVESPINM - CICS PPI initialization member" on page 139.

Following is an example of the information contained in the EVENTASK program-to-program interface initialization member:

```

BUFFQL=20
SERVER=REQUEST,LMT,AUTCPPI,EVEEYPPS
SERVER=RESPONSE,CEMT,AUTCPPI,EVESNRSP
SERVER=RESPONSE,LMT,AUTCPPI,EVESNRSP
SERVER=REQUEST,NACK,AUTCPPI,EVESNACK
SERVER=RESPONSE,NACK,AUTCPPI,EVESNACK
RECEIVERID=NETVCPPI

```

Keyword and Parameter Definitions

BUFFQL

This is a 2 or 3 digit numeric value. The minimum value is 10 and the maximum is 999. If this entry is omitted, a value of 15 is assumed.

SERVER=

These entries define:

1. Whether this function is a REQUEST or a RESPONSE. A REQUEST is used to identify a receiver program to be invoked if NetView gets a CONVERSE or SEND from CICS. A RESPONSE is used to identify a sender program to be invoked if CICS sends a RESPONSE. See "EVESCCCI - CICS to NetView Communication Interface" on page 213.
2. The function, such as LMT (link monitor) or CEMT.
3. The operator ID under which the program runs, for example, AUTCPPI.
4. The command list or command processor used for this function, such as EVEEYPPS (the receiver program for LMT functions from CICS) and EVESNRSP (the common response handler).

RECEIVERID=

The program-to-program interface receiver identifier for the NetView side program-to-program interface subtask program. If omitted, NETVCPPI is assumed.

Comments and Usage Notes

1. A function name may not start with EVE.
2. At least one valid SERVER must be specified.
3. There must be a SERVER entry for each function that uses the CICS Automation program-to-program interface. The corresponding CICS side initialization member entry looks like this:

```
EVEMPINM TYPE=ENTRY,      DEFINE A FUNCTION
      FUNCTION=LMT,        FUNCTION NAME
      TRANSID=COLR         TRANSACTION NAME
```

4. If you are running CICS Automation in more than one NetView domain on the same MVS system, then you need to provide unique RECIEVERIDs in this member. This must also be changed in the corresponding CICS program-to-program interface initialization member EVESPINM. See "EVESPINM - CICS PPI initialization member" on page 139.
5. See "EVENTASK—PPI Initialization Member (NetView)" on page 296 for a sample of this member.

EVESCMT2 - Message exit table for CICS Version 2

Message automation works with messages sent to an MVS console. Some CICS messages are not sent to an MVS console, but to transient data queues instead. The purpose of EVESCMT2 is to identify which of these messages require automation so that they can be sent to an MVS console for automation to take place.

A macro, EVEMCMTE, is used to do this. To define the messages to be automated, edit EVESCMT2 and use the following format:

```
EVEMCMTE TYPE=ENTRY
          ,QUEUE=queue
          ,MSGID=msgid
```

Keyword and Parameter Definitions

QUEUE=

The indirect queue name to which this message is sent.

MSGID=

The message. Asterisks (*) can be used for groups of messages, as in DFH223*.

Comments and Usage Notes

1. The minimum CICS level is 2.1.2.
2. The queue specifications for the EVEMCMTE macros must contain the indirect queue name for the CADL, CSMT, and CSTL destinations.
3. Check your Destination Control Table (DCT) to make sure that these are the correct transient data queue names for these destinations. If they are different, then use your DCT names for your EVEMCMTE definitions.
4. If you are running at CICS Version 3 or higher, refer to “EVESCMT3 - Message exit table for CICS Version 3 and higher” on page 145.
5. EVESCMT2 must be link edited together with the EVESCME2 exit program in one of the CICS DFHRPL libraries. Refer to “EVESJ020—Assemble and Link Edit Definition Members” on page 224.
6. A sample EVESCMT2 member is provided. You will see in this member that EVEMCMTE has several TYPEs:

```
EVEMCMTE TYPE=INITIAL
EVEMCMTE TYPE=ENTRY
EVEMCMTE TYPE=FINAL
EVEMCMTE TYPE=DSECT
```

The only TYPE that you should work with is ENTRY. Do not disturb the other TYPEs. Refer to “EVESCMT2—Message Exit Table for CICS Version 2” on page 267.

Examples of Usage

Example 1

The following is used for VTAM ACB messages:

```
EVEMCMTE TYPE=ENTRY,  
          QUEUE=CSTL,  
          MSGID=DFH3463*
```

Example 2

The following is used for program and transaction abend messages:

```
EVEMCMTE TYPE=ENTRY,  
          QUEUE=CSMT,  
          MSGID=DFH223*
```

```
EVEMCMTE TYPE=ENTRY,  
          QUEUE=CSMT,  
          MSGID=DFH224*
```

Example 3

The following is used for autoinstall error messages:

```
EVEMCMTE TYPE=ENTRY,  
          QUEUE=CSMT,  
          MSGID=DFH3482
```

```
EVEMCMTE TYPE=ENTRY,  
          QUEUE=CSMT,  
          MSGID=DFH3483
```

```
EVEMCMTE TYPE=ENTRY,  
          QUEUE=CADL,  
          MSGID=DFH5935I
```

```
EVEMCMTE TYPE=ENTRY,  
          QUEUE=CADL,  
          MSGID=DFH5966I
```

EVESCMT3 - Message exit table for CICS Version 3 and higher

Message automation works with messages sent to an MVS console. Some CICS messages are not sent to an MVS console, but to transient data queues instead. The purpose of EVESCMT3 is to identify which of these messages require automation so that they can be sent to an MVS console for automation to take place.

A macro, EVEMXMET, is used to do this. To define the messages to be automated, edit EVESCMT3 and use the following EVEMXMET macro format:

```
EVEMXMET TYPE=ENTRY
        ,QUEUE=queue
        ,MSGID=msgid
```

Keyword and Parameter Definitions

QUEUE=

The actual queue name to which this message is sent.

MSGID=

The message ID (without the A, I, or W suffix).

Comments and Usage Notes

1. The queue specifications for the EVEMXMET macros must contain the actual queue names.
2. If you are running at CICS Version 2, refer to “EVESCMT2 - Message exit table for CICS Version 2” on page 143.
3. EVESCMT3 must be link edited together with the EVESCME3 exit program in one of the CICS DFHRPL libraries. Refer to “EVESJ020—Assemble and Link Edit Definition Members” on page 224.
4. A sample EVESCMT3 member is provided. You will see in this member that EVEMXMET has several TYPEs:

```
EVEMXMET TYPE=INITIAL
EVEMXMET TYPE=ENTRY
EVEMXMET TYPE=FINAL
EVEMXMET TYPE=DSECT
```

The only TYPE that you should work with is ENTRY. Do not disturb the other TYPEs. Refer to “EVESCMT3—Message Exit Table for CICS Version 3 and Higher” on page 271.

Examples of Usage

Example 1

The following is used for VTAM ACB messages:

```
EVEMXMET TYPE=ENTRY,  
          QUEUE=CSNE,  
          MSGID=DFHZC3463
```

Example 2

The following is used for program and transaction abend messages:

```
EVEMXMET TYPE=ENTRY,  
          QUEUE=CSMT,  
          MSGID=DFHAC2230
```

```
EVEMXMET TYPE=ENTRY,  
          QUEUE=CSMT,  
          MSGID=DFHAC2236
```

```
EVEMXMET TYPE=ENTRY,  
          QUEUE=CSMT,  
          MSGID=DFHAC2237
```

```
EVEMXMET TYPE=ENTRY,  
          QUEUE=CSMT,  
          MSGID=DFHAC2238
```

```
EVEMXMET TYPE=ENTRY,  
          QUEUE=CSMT,  
          MSGID=DFHAC2240
```

```
EVEMXMET TYPE=ENTRY,  
          QUEUE=CSMT,  
          MSGID=DFHAC2241
```

```
EVEMXMET TYPE=ENTRY,  
          QUEUE=CSMT,  
          MSGID=DFHAC2242
```

```
EVEMXMET TYPE=ENTRY,  
          QUEUE=CSMT,  
          MSGID=DFHAC2243
```

Example 3

The following is used for autoinstall error messages:

```
EVEMXMET TYPE=ENTRY,  
  QUEUE=CSNE,  
  MSGID=DFHZC3482
```

```
EVEMXMET TYPE=ENTRY,  
  QUEUE=CSNE,  
  MSGID=DFHZC3483
```

```
EVEMXMET TYPE=ENTRY,  
  QUEUE=CADL,  
  MSGID=DFHZC5966
```

Part 3. Planning, Installation, and Customization

Chapter 5. Planning

This chapter describes the steps you will take to plan for installation, customization, and implementation of CICS Automation.

Follow each step in the order in which it is given and use the check lists to track your progress. Some of the check lists are used multiple times. Extra copies are provided in Appendix F, “Extra Check Lists” on page 331.

Pre-Installation and Planning Steps

Complete the steps listed here before installing the CICS Automation code. Use this table to keep track of your completed work:

<i>Table 13. Planning Check List</i>	
√	Step
_____	1: Verify that SA OS/390 on each NetView domain in which CICS 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 are going to 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: Fill out the worksheets. See “Designing Your Automation Environment” on page 152.
_____	4: Schedule CICS subsystem initializations, NetView initializations, and system IPLs according to the directions in the worksheets shown in “System IPLs, NetView Restarts, and CICS Restarts” on page 327.

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 CICS region is defined to CICS Automation as a subsystem. There are a set of control file entries for each CICS region. 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 CICS regions by NetView domain. In this step, you will determine the names that you will use to identify each CICS region to CICS Automation. Use the worksheet in “NetView Domains and CICS Regions” on page 307.
2. Gather basic automation information for each CICS subsystem. Use a separate worksheet for each NetView domain. See “Basic CICS Subsystem Information Sheets” on page 309.
3. List the IDs of the CICS Automation automation operators. Use the worksheets in “Automation Operator Information Sheets” on page 312.

The following steps are optional.

4. If you are going to group CICS subsystems so that they can be started or shut down from the operator interface in groups, fill out the worksheets in “CICS Subsystem Group Information Sheet” on page 315.

Note: You must have single-point-of-control defined to be able to start up or shut down subsystems across domains.

5. If you are going to use service periods, fill out the worksheets in “Service Periods Information Sheets” on page 317.
6. If you are going to use link monitoring, fill out the worksheets in “Link Monitoring Information Sheets” on page 319.
7. If you are going to use triggers, fill out the worksheets in “Startup and Shutdown Trigger Information Sheets” on page 322.
8. If you are going to use health checking, fill out the worksheets in “Health Checking Information Sheets” on page 325.

Chapter 6. Installation

Complete the steps listed here. Use the following table to track your completed work:

Table 14. Installation Check List	
√	Step
_____	1: Using the information in the CICS Automation program directory, load the CICS Automation libraries from the distribution tape.
_____	2: Add the following entry in MPFLSTxx in SYS1.PARMLIB to trap all DFH and EVE prefix messages: DFH*,SUP(NO),AUTO(YES) EVE*,SUP(NO),AUTO(YES) This CICS Automation requirement forwards these messages to NetView.
_____	3: Update IEAAPFxx in SYS1.PARMLIB to provide APF authorization.
_____	4. Define subsystem allocatable consoles.
_____	5. Check the subsystem name table to ensure that the NetView SSI is first.

Note: The AOC CICS feature uses messages IEF403I and IEF404I to track a subsystem. These messages are issued only when jobname monitoring has been requested. Therefore, be sure that MONITOR JOBNAME,S,T has been issued. This command can be included in your COMMNDxx parmlib member.

Step 3: Updating 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 (IEAAPFxx) of SYS1.PARMLIB. After IEAAPFxx is updated, MVS must be re-IPLed (unless you are using dynamic allocation in MVS 4.3 or higher).

Ensure that the AOCCICS.V1R4M0.SEVEMOD1 library used when link-editing AOC CICS Automation is authorized. If you specify an unauthorized library on a STEPLIB or concatenate unauthorized libraries with authorized libraries, all libraries are treated as if they are unauthorized.

Where to get more information

More information on APF authorization may be found in *MVS/ESA Initialization and Tuning*.

Step 4: Defining Subsystem Allocatable Consoles

The NetView program's use of the MVS subsystem interface allows MVS system operator commands to be issued 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 CICS automation operator can use a subsystem allocatable console. Make sure the number of subsystem allocatable consoles is increased by the number of CICS automation operators.

Where to get more information

Information on defining subsystem allocatable consoles for your system is in *MVS/ESA Installation: System Generation Reference*. Also see *MVS/ESA Input Output Configuration Program User's Guide and Reference*, or refer to *MVS/ESA Initialization and Tuning*, whichever is appropriate for your system.

Step 5: Check the Subsystem Name Table

The first active NetView SSI is used for program-to-program interface communication. If an SSI for a NetView other than the one running CICS Automation is higher in the table, then that SSI will be used for the program-to-program interface, disrupting CICS Automation program-to-program interface communications.

Check the subsystem name table in MVS SYS1.PARMLIB, member IEFSSNxx to verify that the NetView SSI that is used by CICS Automation is first in the list (ahead of all other NetView subsystem names).

Note: Non-NetView subsystems, such as JES2 may, however, precede this entry.

Use an option available with NetView Version 2 Release 3 or later to specify "NOPPI" on all NetView SSIs except the SSI that CICS Automation uses. This "NOPPI" option is specified as a startup parameter on the SSI JCL.

Chapter 7. Merge NetView Related Members

This section describes how to build the CICS Automation parameter data sets and assemble the code that enables CICS Automation to operate in the NetView environment.

Use the following table to track your completed work:

<i>Table 15. Merging the NetView Related Members Check List</i>	
✓	Step
_____	1: Add CICS Automation data sets to NetView JCL
_____	2: Copy CICS Automation sample members to the target library
_____	3: Merge Status Display Facility members
_____	4: Merge EVECFG into the control file
_____	5: Merge EVECMD into DSICMD
_____	6: Merge the NetView Automation Table
_____	7: Merge for CDRM recovery
_____	8: Merge EVEOPF into DSIOPF
_____	9: Merge the NetView profile data set
_____	10: Merge EVEDMN into DSIDMN

Step 1: Add CICS Automation Data Sets to NetView JCL

Add the following libraries to your NetView JCL procedure. Review the NetView JCL procedure to verify that the CICS 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=...).

DD	DSN	Description
DSICLD	AOCCICS.V1R4M0.SEVENCL1	Command lists
CNMPNL1	AOCCICS.V1R4M0.SEVENPN1	Panels
CNMPNL1	AOCCICS.V1R4M0.SEVENPN2	Message on-line help
DSIMSG	AOCCICS.V1R4M0.SEVENMSG	Communication messages
STEPLIB	AOCCICS.V1R4M0.SEVEMOD1	Authorized NetView user link library containing the linked CICS Automation modules.

Note: For performance reasons, these data sets should be high in the concatenation chain.

Step 2: Copy CICS Automation Sample Members to the Target Library

Do one of the following:

- Run job EVESJ010 to copy from the sample library 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 procedure.

Step 3: Merge Status Display Facility Members

To merge these members:

1. Merge EVETREE from the sample library (AOCCICS.V1R4M0.SEVESAMP) into the existing AOFTREE tree structure member. EVETREE contains the tree structure for the CICS Automation Status Display Facility panels.
Note: If you are using the %INCLUDE facility for multiple tree structures, EVETREE must be copied into each.
2. Merge EVEPNLS from the sample library (AOCCICS.V1R4M0.SEVESAMP) into the existing Status Display Facility panel definition member. EVEPNLS contains a list of %INCLUDE statements for all of the Status Display Facility panels used with CICS Automation.

Step 4: Include EVECFG into the Control File

%INCLUDE EVECFG from the CICS Automation sample library (AOCCICS.V1R4M0.SEVESAMP) into the running SA OS/390 control file. (This member was already copied from the CICS Automation sample library in Step 2.) This member contains the %INCLUDE statement that causes member EVECFG01 to be processed by SA OS/390.

Step 5: Include EVECMD into DSICMD

%INCLUDE EVECMD from the CICS Automation sample library (AOCCICS.V1R4M0.SEVESAMP) into DSICMD before the END statement. (This member was already copied from the CICS Automation sample library in Step 2.) EVECMD contains the additional commands required for CICS Automation.

Step 6: Include and Update the NetView Automation Table

To include and update the automation table:

- ___ 1. %INCLUDE CICSMSG0 into your NetView/SA OS/390 message table.

 CICSMSG0 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 CICSMSG0

 after the %INCLUDE for AOFMSGSY in member AOFMSG00. Please note that CICSMSG0 contains %INCLUDE statements for a number of other members that include:
 - EVEMCON0 — 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.
 - EVEMEVE0 — EVE messages that are used by CICS automation
- ___ 2. %INCLUDE CICSMSG1 into your NetView/SA OS/390 message table.

 CICSMSG1 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 CICSMSG1

 after the %INCLUDE for AOFMSGSY in member AOFMSG01. Please note that CICSMSG1 contains %INCLUDE statements for a number of other members that include:
 - EVEMCON1 — 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.
 - EVEMDFH1 — CICS/ESA messages used by CICS Automation
 - EVEMEVE1 — EVE messages that are used by CICS Automation

Step 7: Tailoring Your Network NetView for CDRM Recovery

If you have a separate network NetView, then do the following:

- ___ 1. Add the following to the network NetView Automation Table:

 IF MSGID='IST093I' & TOKEN(2)=RESOURCE .
 THEN EXEC(CMD('EVEEZ000 ' RESOURCE) ROUTE(ALL *));
- ___ 2. Copy exec EVEEZ000 from AOCCICS.V1R4M0.SEVENCL1 into any command list library (DSICLD) that is in your NetView network chain.

Warning

The network NetView Automation Table is used if two copies of NetView are running in the same system, one for console automation and one for network automation.

Step 8: Include EVEOPF into DSIOPF

To include this member:

- ___ 1. %INCLUDE member EVEOPF from the CICS Automation sample library into DSIOPF. (This member was already copied from the CICS Automation sample library in Step 2.)
- ___ 2. Verify that the autotask names are correct according to the worksheet shown in “Automation Operator Information Sheets” on page 312. This worksheet was filled out during the planning phase.

EVEOPF is used to define the CICS Automation autotasks.

Step 9: Merge the NetView Profile Data Set

This is the profile for the automated operators. Do one of the following:

- Copy EVEPRFAO from the CICS Automation sample library (AOCCICS.V1R4M0.SEVISAMP) into your NetView profile (DSIPRF).
- Concatenate the sample library to the DSIPARM data set in the NetView Startup procedure.

Step 10: Include EVEDMN into DSIDMN and Update

The purpose of EVEDMN is to add the EVENTASK to DSIDMN. To merge this member:

- ___ 1. %INCLUDE EVEDMN from the CICS Automation sample library into DSIDMN. (This member was already copied from the CICS Automation sample library in Step 2.)
- ___ 2. Calculate the number of CICS Automation autotasks required for this domain. Add that number to the POSPOOL statement in DSIDMN. This number must be at least the number of autotasks added in Step 8.
- ___ 3. In your VTAM definitions, verify that there are enough VTAM APPL statements to allow all of the operators to be logged on.

Chapter 8. Define CICS Automation to CICS

This chapter describes the basic CICS Automation definitions that take place on CICS. Because of the differences in CICS versions and releases, we attempt to avoid providing installation or definition instructions specific to CICS. Therefore, you should refer to the CICS documentation while performing these steps, especially the following:

- Either the *CICS/MVS Resource Definition (Macro)* or the *CICS/ESA Resource Definition (Macro)* publication.
- Either the *CICS/MVS Resource Definition (Online)* or the *CICS/ESA Resource Definition (Online)* publication.

A helpful note

These steps are performed on each CICS region. Additional check lists are provided in Appendix F, “Extra Check Lists” on page 331.

Use the following table to track your completed work:

Table 16. CICS Definitions Check List	
√	Step
_____	1: SIT or startup overrides
_____	2: Program list table definitions
_____	3: Transaction list table definitions
_____	4: Define consoles
_____	5: Transaction and program definitions
_____	6: DFHRPL and the CICS Automation library

Note: The TS queues EVEVCQUE and COLLEEVEQ used by SA OS/390 CICS must not be defined as remote in your TST (temporary storage table).

Step 1: SIT or Startup Overrides

1. On each CICS, ensure that the system initialization table or startup overrides include the following:
PLTPI=xx, where xx is the suffix to the startup PLT
PLTSD=yy, where yy is the suffix to the shutdown PLT
EXEC=YES,
EXITS=YES,
MSGLVL=1,
BMS=(STANDARD|FULL),
XLT=zz where zz is the suffix to the transaction list tables

Note: **XLT=zz** is not required for CICS 4.1 and higher.

2. Because CICS Automation maintains a long running task in each CICS, review the AMXT, CMXT, and MXT values.

- ___ 3. You must add 'CN' as your last startup override, whether from SYSIN or through the JCL, in order to support the CICS Automation operator interface SIT override option.

Step 2: Program List Table Definitions

- ___ 1. Add the TYPE=ENTRY definitions shown in the following example to the post initialization program list table (PLT) for each CICS after the entry for DFHDELIM (as in phase 2).

```
DFHPLT TYPE=INITIAL,SUFFIX=xx
DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM Only required
for CICS Version 3
DFHPLT TYPE=ENTRY,PROGRAM=EVESTIEX See warning below
DFHPLT TYPE=ENTRY,PROGRAM=EVESTISP
DFHPLT TYPE=FINAL
```

Warning about EVESTIEX!

This program enables message exit processing and is only available with CICS Version 2.1.2 and above or CICS Version 3.2 and above. **Do not** add this entry if you are not at these levels.

- ___ 2. Add the TYPE=ENTRY definitions shown in the following example to the shut down program list table (PLT) for each CICS.

```
DFHPLT TYPE=INITIAL,SUFFIX=yy
DFHPLT TYPE=ENTRY,PROGRAM=EVESPLTT
DFHPLT TYPE=ENTRY,PROGRAM=EVESYLMQ
DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
DFHPLT TYPE=FINAL
```

- ___ 3. Assemble the PLT table.
- ___ 4. If you must define a new table, add the following definitions:

```
DEFINE PROGRAM(DFHPLTxx) LANGUAGE(ASSEMBLER) GROUP(yourgroup)
DEFINE PROGRAM(DFHPLTyy) LANGUAGE(ASSEMBLER) GROUP(yourgroup)
```

Step 3: Transaction List Table Definitions

Note: This step is not required for CICS 4.1 and higher.

- ___ 1. Add the following to the transaction list table (XLT) for each CICS. This enables transaction COMC to process during a CICS shutdown:

```
DFHXLT TYPE=INITIAL,SUFFIX=zz
DFHXLT TYPE=ENTRY,TRANSID=COMC
DFHXLT TYPE=FINAL
```

- ___ 2. Assemble the XLT table.
 - ___ 3. If you must define a new table, add the following definitions:
- ```
DEFINE PROGRAM(DFHXLTzz) LANGUAGE(ASSEMBLER) GROUP(yourgroup)
```

---

## Step 4: Define Consoles

- \_\_\_ 1. Define a console with either a CONSOLE number of 00 or a CONSNAME of INTERNAL (the CONSOLE and CONSNAME parameters are mutually exclusive). This enables transactions COPS (start program-to-program interface) and COPP (stopprogram-to-program interface). Verify that the SAF security profile associated with console 00 has sufficient authority to issue the COPP and COPS transactions.
- \_\_\_ 2. Define consoles for autotasks to enable shutdown and broadcast functions.

In an EMCS environment the autotask console names are determined, in order of precedence as follows:

- a. If you are using AOCGETCN (that is, using the profiles shipped with the product) the name is determined by AOFCNMASK. For more information, see *AOC/MVS Customization and Programming or System Automation for OS/390 Customization*.
- b. If you are using NetView 3.1 or above, the CONSNAME parameter on the PROFILE statement in the task profile determines the EMCS console name. For more information, see *NetView for MVS V3 Administration and Security Reference*.
- c. By default the autotask name is used for the EMCS console name.

SHUTOPER determines the autotask name used for shutdown; the autotasks defined for the CICS AUTOOPS are used for broadcast.

In a MCS environment, define consoles with CONSOLE numbers from 01 to nn for each subsystem-allocatable console.

---

## Step 5: Transaction and Program Definitions

This step describes how to define the standard CICS Automation transactions and programs to CICS. Two methods can be used:

### Method 1: Batch

Uses the DFHCSDUP program. You must be at CICS 2.1.2 or CICS 3.2 or higher to use this method. Otherwise, use method 2.

### Method 2: Online

Lists each of the transactions and programs so that they can be defined manually.

Use whichever method is preferable.

### Notes:

- 1. You may need to add other parameters, such as transaction class to the above.
- 2. After making the definitions, INSTALL the group or groups that contain the definitions.
- 3. If you are using PPT and PCT definitions for your CICS regions, add equivalent entries to those tables.

## Method 1: Batch

The members required to run these jobs are provided with CICS Automation. However, some modifications are required, as described below:

### A helpful hint

You might want to backup your CSDs before doing this step.

For each CSD, run the job shown in “EVESJ015—CICS/MVS Definitions” on page 223. This job defines transactions and programs for CICS Automation in three groups—EVEGRP1, EVEGRP2, and EVEGRP3. After running this job, you will need to INSTALL these three groups on each CICS.

Modify this job as follows:

```
//STEPLIB DD DISP=SHR,DSN=CICSxxy.LOADLIB (See note 1.)
// DD DISP=SHR,DSN=AOC CICS.V1R4M0.SEVEMOD2 (See note 2.)
//DFHCSD DD DISP=SHR,DSN=yourCSD.DFHCSD (See note 3.)
//SYSUT1 DD UNIT=3380,SPACE=(1024,(100,100))
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
 UPGRADE USING(EVECSxx) (See note 4.)
 ADD GROUP(EVEGRP1) LIST(yourlist) (See note 5.)
 ADD GROUP(EVEGRP2) LIST(yourlist)
 ADD GROUP(EVEGRP3) LIST(yourlist)
/*
//
```

### Notes:

1. Change the data set name of this file to match your CICS load library, where *xx* is 21, 32, 33, 41, 51 or 52, depending on your level of CICS. 51 refers to CICS TS for OS/390 R1 and 52 refers to CICS TS for OS/390 R2.
2. If required, change the data set name on this file to match your CICS Automation load library.
3. Change the data set name to match your CICS CSD for this CICS.
4. Change *xx* to 21, 32, 33, or 41, depending on your CICS level.
5. Change *xx* to 21, 32, 33, 41, 51 or 52, depending on your CICS level. 51 refers to CICS TS for OS/390 R1 and 52 refers to CICS TS for OS/390 R2.
6. Change *yourlist* for EVEGRP1, EVEGRP2, and EVEGRP3 to the name of the group lists for this CICS.

## Method 2: Online

There are some differences depending on the level of CICS. Be sure that your level supports the parameters used in these definitions, such as the DESCRIPTION parameter shown in the CICS/ESA definitions.

**If this is CICS/MVS, add the following definitions:**

```
_____ DEFINE TRANSACTION(COHO) PROGRAM(EVESECHO) GROUP(EVEGRP1)
_____ DEFINE TRANSACTION(COPC) PROGRAM(EVESPPIC) GROUP(EVEGRP1)
_____ DEFINE TRANSACTION(COPP) PROGRAM(EVESPPIP) GROUP(EVEGRP1)
_____ DEFINE TRANSACTION(COPS) PROGRAM(EVESPPIS) GROUP(EVEGRP1)
_____ DEFINE TRANSACTION(COMT) PROGRAM(EVESECMT) TWASIZE(512) GROUP(EVEGRP1)
_____ DEFINE TRANSACTION(COMC) PROGRAM(EVESCOMC) TWASIZE(512) GROUP(EVEGRP1)
_____ DEFINE TRANSACTION(COHR) PROGRAM(EVESHLTH) GROUP(EVEGRP1)
_____ DEFINE PROGRAM(EVESTISP) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
_____ DEFINE PROGRAM(EVESTIEX) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
_____ DEFINE PROGRAM(EVESECHO) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
_____ DEFINE PROGRAM(EVESCCCI) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
_____ DEFINE PROGRAM(EVESCME2) LANGUAGE(ASSEMBLER) RELOAD(NO) GROUP(EVEGRP1)
_____ DEFINE PROGRAM(EVESPERR) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
_____ DEFINE PROGRAM(EVESPMMSG) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
_____ DEFINE PROGRAM(EVESPPIP) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
_____ DEFINE PROGRAM(EVESPPI3) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
_____ DEFINE PROGRAM(EVESECMT) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
_____ DEFINE PROGRAM(EVESCOMC) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
_____ DEFINE PROGRAM(EVESPINM) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
_____ DEFINE PROGRAM(EVESPLTT) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
_____ DEFINE PROGRAM(EVESPPIC) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
_____ DEFINE PROGRAM(EVESPPIS) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
_____ DEFINE PROGRAM(EVESHLTH) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
_____ DEFINE TRANSACTION(COLO) PROGRAM(EVESYLOI) RSLC(NO) GROUP(EVEGRP2)
_____ DEFINE TRANSACTION(COLR) PROGRAM(EVESYPPS) GROUP(EVEGRP2)
```

```

_____ DEFINE TRANSACTION(COLC) PROGRAM(EVESYLM I) GROUP(EVEGRP2)
_____ DEFINE TRANSACTION(COLE) PROGRAM(EVESYLM E) GROUP(EVEGRP2)
_____ DEFINE PROGRAM(EVESYPP I) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
_____ DEFINE PROGRAM(EVESYPPS) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
_____ DEFINE PROGRAM(EVESYLRQ) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
_____ DEFINE PROGRAM(EVESYLOI) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
_____ DEFINE PROGRAM(EVESYLS D) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
_____ DEFINE PROGRAM(EVESYLD T) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
_____ DEFINE PROGRAM(EVESYLS U) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
_____ DEFINE PROGRAM(EVESYLS P) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
_____ DEFINE PROGRAM(EVESYS P S) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
_____ DEFINE PROGRAM(EVESYS P O) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
_____ DEFINE PROGRAM(EVESYHLP) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
_____ DEFINE PROGRAM(EVESYLM I) LANGUAGE(ASSEMBLER) RELOAD(NO) GROUP(EVEGRP2)
_____ DEFINE PROGRAM(EVESYLM) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
_____ DEFINE PROGRAM(EVESYLM A) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
_____ DEFINE PROGRAM(EVESYLM E) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
_____ DEFINE PROGRAM(EVESYLM O) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
_____ DEFINE PROGRAM(EVESYLM R) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
_____ DEFINE PROGRAM(EVESYLM S) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
_____ DEFINE PROGRAM(EVESYLM T) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
_____ DEFINE PROGRAM(EVESYLM Q) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
_____ DEFINE MAPSET(EVEM100) GROUP(EVEGRP3)
_____ DEFINE MAPSET(EVEM130) GROUP(EVEGRP3)
_____ DEFINE MAPSET(EVEM135) GROUP(EVEGRP3)
_____ DEFINE MAPSET(EVEM136) GROUP(EVEGRP3)
_____ DEFINE MAPSET(EVEM140) GROUP(EVEGRP3)
_____ DEFINE MAPSET(EVEM150) GROUP(EVEGRP3)
_____ DEFINE MAPSET(EVEM151) GROUP(EVEGRP3)

```



```
____ DEFINE MAPSET(EVEM152) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEM153) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH100) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH101) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH130) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH131) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH135) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH136) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH137) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH140) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH150) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH151) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH152) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH153) GROUP(EVEGRP3)
```

**If this is CICS/ESA 3.2, add the following definitions:**

```
____ DEFINE TRANSACTION(COHO) PROGRAM(EVESECHO) GROUP(EVEGRP1) DESCRIPTION(CICSAO PPI transaction)
____ DEFINE TRANSACTION(COPC) PROGRAM(EVESPPIC) GROUP(EVEGRP1) DESCRIPTION(CICSAO PPI transaction)
____ DEFINE TRANSACTION(COPP) PROGRAM(EVESPPIP) GROUP(EVEGRP1) DESCRIPTION(CICSAO stop PPI transaction)
____ DEFINE TRANSACTION(COPS) PROGRAM(EVESPPIS) GROUP(EVEGRP1) DESCRIPTION(CICSAO start PPI transaction)
____ DEFINE TRANSACTION(COMC) PROGRAM(EVESCOMC) TWASIZE(512) DESCRIPTION(CICSAO CEMT Interface Transaction) GROUP(EVEGRP1)
____ DEFINE TRANSACTION(COMT) PROGRAM(EVESCMT) TWASIZE(512) DESCRIPTION(CICSAO CEMT Interface Transaction) GROUP(EVEGRP1)
____ DEFINE TRANSACTION(COHR) PROGRAM(EVESHLTH) DESCRIPTION(CICSAO health interface transaction) GROUP(EVEGRP1)
____ DEFINE PROGRAM(EVESTISP) LANGUAGE(ASSEMBLER) DESCRIPTION(CICSAO enable PPI) GROUP(EVEGRP1)
____ DEFINE PROGRAM(EVESTIEX) LANGUAGE(ASSEMBLER) DESCRIPTION(CICSAO enable exits) GROUP(EVEGRP1)
____ DEFINE PROGRAM(EVESECHO) LANGUAGE(ASSEMBLER) DESCRIPTION(CICSAO Console Echo program) GROUP(EVEGRP1)
____ DEFINE PROGRAM(EVESCCCI) LANGUAGE(ASSEMBLER) DESCRIPTION(CICSAO PPI Interface Program) GROUP(EVEGRP1)
____ DEFINE PROGRAM(EVESCME3) LANGUAGE(ASSEMBLER) RELOAD(NO) DESCRIPTION(CICSAO PPI XMEOUT user exit) GROUP(EVEGRP1)
____ DEFINE PROGRAM(EVESPERR) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
____ DEFINE PROGRAM(EVESMSG) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1) DESCRIPTION(CICSAO message handling program)
____ DEFINE PROGRAM(EVESPPIP) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1) DESCRIPTION(CICSAO stop PPI program)
____ DEFINE PROGRAM(EVESPPI3) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1) DESCRIPTION(CICSAO start PPI program)
____ DEFINE PROGRAM(EVESHLTH) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1) DESCRIPTION(CICSAO health interface program)
____ DEFINE PROGRAM(EVESCMT) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1) DESCRIPTION(CICSAO CEMT interface program)
____ DEFINE PROGRAM(EVESCOMC) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1) DESCRIPTION(CICSAO CEMT interface program)
____ DEFINE PROGRAM(EVESPINM) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1) DESCRIPTION(CICSAO PPI function table)
____ DEFINE PROGRAM(EVESPLTT) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1) DESCRIPTION(CICSAO PLTSD program)
____ DEFINE PROGRAM(EVESPPIC) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1) DESCRIPTION(CICSAO PPI interface program)
____ DEFINE PROGRAM(EVESPPIS) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1) DESCRIPTION(CICSAO start PPI program)
____ DEFINE TRANSACTION(COLO) PROGRAM(EVESYLOI) DESCRIPTION(CICSAO LMT Operator Interface) GROUP(EVEGRP2)
____ DEFINE TRANSACTION(COLR) PROGRAM(EVESYPPS) GROUP(EVEGRP2)
____ DEFINE TRANSACTION(COLC) PROGRAM(EVESYLM I) GROUP(EVEGRP2)
____ DEFINE TRANSACTION(COLE) PROGRAM(EVESYLME) GROUP(EVEGRP2)
____ DEFINE PROGRAM(EVESYPI) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DESCRIPTION(CICSAO LMT PPI program)
____ DEFINE PROGRAM(EVESYPPS) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DESCRIPTION(CICSAO LMT PPI program)
____ DEFINE PROGRAM(EVESYLRQ) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DESCRIPTION(CICSAO LMT requestor)
____ DEFINE PROGRAM(EVESYLOI) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DESCRIPTION(CICSAO LMT requestor)
____ DEFINE PROGRAM(EVESYLS D) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DESCRIPTION(CICSAO LMT main menu)
____ DEFINE PROGRAM(EVESYLD T) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DESCRIPTION(CICSAO LMT Operator interface)
____ DEFINE PROGRAM(EVESYLU S) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DESCRIPTION(CICSAO LMT Operator interface)
____ DEFINE PROGRAM(EVESYLS P) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DESCRIPTION(CICSAO LMT program)
```

```

____ DEFINE PROGRAM(EVESYSPS) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DESCRIPTION(CICSAO LMT program)
____ DEFINE PROGRAM(EVESYSPO) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DESCRIPTION(CICSAO LMT program)
____ DEFINE PROGRAM(EVESYHLP) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DESCRIPTION(CICSAO LMT help)
____ DEFINE PROGRAM(EVESYLM) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DESCRIPTION(CICSAO LMT program)
____ DEFINE PROGRAM(EVESYLMQ) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DESCRIPTION(CICSAO LMT program)
____ DEFINE PROGRAM(EVESYLME) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DESCRIPTION(CICSAO LMT program)
____ DEFINE PROGRAM(EVESYLMR) LANGUAGE(ASSEMBLER) RELOAD(NO) DESCRIPTION(CICSAO LMT program) GROUP(EVEGRP2)
____ DEFINE PROGRAM(EVESYLMQ) LANGUAGE(ASSEMBLER) DESCRIPTION(CICSAO LMT program) GROUP(EVEGRP2)
____ DEFINE PROGRAM(EVESYLMQ) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DESCRIPTION(CICSAO LMT program)
____ DEFINE PROGRAM(EVESYLMR) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DESCRIPTION(CICSAO LMT program)
____ DEFINE PROGRAM(EVESYLS) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DESCRIPTION(CICSAO LMT program)
____ DEFINE PROGRAM(EVESYLS) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DESCRIPTION(CICSAO LMT program)
____ DEFINE PROGRAM(EVESYLS) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DESCRIPTION(CICSAO LMT program)
____ DEFINE MAPSET(EVEM100) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEM130) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEM135) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEM136) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEM140) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEM150) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEM151) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEM152) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEM153) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH100) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH101) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH130) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH131) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH135) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH136) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH137) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH140) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH150) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH151) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH152) GROUP(EVEGRP3)
____ DEFINE MAPSET(EVEH153) GROUP(EVEGRP3)

```

**If this is CICS/ESA 3.3, add the following definitions:**

```
____ DEFINE TRANSACTION(COHO) PROGRAM(EVESECHO) GROUP(EVEGRP1) TASKDATALOC(ANY) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO PPI transaction)

____ DEFINE TRANSACTION(COPC) PROGRAM(EVESPPIC) GROUP(EVEGRP1) TASKDATALOC(ANY) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO PPI transaction)

____ DEFINE TRANSACTION(COPP) PROGRAM(EVESPPIP) GROUP(EVEGRP1) TASKDATALOC(ANY) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO stop PPI transaction)

____ DEFINE TRANSACTION(COPS) PROGRAM(EVESPPIS) GROUP(EVEGRP1) TASKDATALOC(ANY) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO start PPI transaction)

____ DEFINE TRANSACTION(COMC) PROGRAM(EVESCOMC) TWASIZE(512) TASKDATALOC(BELOW) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO CEMT Interface Transaction) GROUP(EVEGRP1)

____ DEFINE TRANSACTION(COMT) PROGRAM(EVESCMT) TWASIZE(512) TASKDATALOC(BELOW) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO CEMT Interface Transaction) GROUP(EVEGRP1)

____ DEFINE TRANSACTION(COHR) PROGRAM(EVESHLTH) GROUP(EVEGRP1) TASKDATALOC(ANY) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO Health Checking Interface)

 *
 * PROGRAMS for EVEGRP1 - Base and PPI
 *

____ DEFINE PROGRAM(EVESTISP) LANGUAGE(ASSEMBLER) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO enable PPI) GROUP(EVEGRP1)

____ DEFINE PROGRAM(EVESTIEX) LANGUAGE(ASSEMBLER) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO enable exits) GROUP(EVEGRP1)

____ DEFINE PROGRAM(EVESECHO) LANGUAGE(ASSEMBLER) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO Console Echo program) GROUP(EVEGRP1)

____ DEFINE PROGRAM(EVESCCCI) LANGUAGE(ASSEMBLER) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO PPI Interface Program) GROUP(EVEGRP1)

____ DEFINE PROGRAM(EVESCM33) LANGUAGE(ASSEMBLER) RELOAD(NO) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO PPI XMEOUT user exit) GROUP(EVEGRP1)

____ DEFINE PROGRAM(EVESPERR) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1) DATALOCATION(ANY) EXECKEY(CICS)

____ DEFINE PROGRAM(EVESPMMSG) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO message handling program)

____ DEFINE PROGRAM(EVESPPIP) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO stop PPI program)

____ DEFINE PROGRAM(EVESPPI3) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO start PPI program)

____ DEFINE PROGRAM(EVESCMT) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1) DATALOCATION(BELOW) EXECKEY(CICS)
 DESCRIPTION(CICS AO CEMT interface program)

____ DEFINE PROGRAM(EVESCOMC) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1) DATALOCATION(BELOW) EXECKEY(CICS)
 DESCRIPTION(CICS AO CEMT interface program)

____ DEFINE PROGRAM(EVESPINM) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO PPI function table)

____ DEFINE PROGRAM(EVESPLTT) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO PLTSD program)

____ DEFINE PROGRAM(EVESPPIC) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO PPI interface program)

____ DEFINE PROGRAM(EVESPPIS) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO start PPI program)

____ DEFINE PROGRAM(EVESHLTH) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1) DATALOCATION(ANY) EXECKEY(CICS)
```

```

 DESCRIPTION(CICS AO Health Checking Interface)

*
* TRANSACTIONS FOR EVEGRP2 - LMT
*

____ DEFINE TRANSACTION(COLO) PROGRAM(EVESYLOI) TASKDATALOC(ANY) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO LMT Operator Interface) GROUP(EVEGRP2)

____ DEFINE TRANSACTION(COLR) PROGRAM(EVESYPPS) GROUP(EVEGRP2) TASKDATALOC(ANY) TASKDATAKEY(USER)

____ DEFINE TRANSACTION(COLC) PROGRAM(EVESYLM I) GROUP(EVEGRP2) TASKDATALOC(BELOW) TASKDATAKEY(USER)

____ DEFINE TRANSACTION(COLE) PROGRAM(EVESYLM E) GROUP(EVEGRP2) TASKDATALOC(ANY) TASKDATAKEY(USER)

*
* programs for EVEGRP2 - LMT
*

____ DEFINE PROGRAM(EVESYPPI) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT PPI program)

____ DEFINE PROGRAM(EVESYPPS) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT PPI program)

____ DEFINE PROGRAM(EVESYLRQ) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT requestor)

____ DEFINE PROGRAM(EVESYLOI) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT requestor)

____ DEFINE PROGRAM(EVESYLS D) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT main menu)

____ DEFINE PROGRAM(EVESYLD T) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT Operator interface)

____ DEFINE PROGRAM(EVESYLS U) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT Operator interface)

____ DEFINE PROGRAM(EVESYLS P) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program)

____ DEFINE PROGRAM(EVESYS P S) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program)

____ DEFINE PROGRAM(EVESYS P O) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program)

____ DEFINE PROGRAM(EVESYHLP) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT help)

____ DEFINE PROGRAM(EVESYLM) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DATALOCATION(BELOW) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program)

____ DEFINE PROGRAM(EVESYLM A) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program)

____ DEFINE PROGRAM(EVESYLM E) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program)

____ DEFINE PROGRAM(EVESYLM I) LANGUAGE(ASSEMBLER) RELOAD(NO) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program) GROUP(EVEGRP2)

____ DEFINE PROGRAM(EVESYLM O) LANGUAGE(ASSEMBLER) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program) GROUP(EVEGRP2)

____ DEFINE PROGRAM(EVESYLM Q) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program)

____ DEFINE PROGRAM(EVESYLM R) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program)

```

```

____ DEFINE PROGRAM(EVESYLS) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program)

____ DEFINE PROGRAM(EVESYLM) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2) DATALOCATION(BELOW) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program)

 *
 * MAPSETS for LMT EUI - EVEGRP3
 *

____ DEFINE MAPSET(EVEM100) GROUP(EVEGRP3)

____ DEFINE MAPSET(EVEM130) GROUP(EVEGRP3)

____ DEFINE MAPSET(EVEM135) GROUP(EVEGRP3)

____ DEFINE MAPSET(EVEM136) GROUP(EVEGRP3)

____ DEFINE MAPSET(EVEM140) GROUP(EVEGRP3)

____ DEFINE MAPSET(EVEM150) GROUP(EVEGRP3)

____ DEFINE MAPSET(EVEM151) GROUP(EVEGRP3)

____ DEFINE MAPSET(EVEM152) GROUP(EVEGRP3)

____ DEFINE MAPSET(EVEM153) GROUP(EVEGRP3)

____ DEFINE MAPSET(EVEH100) GROUP(EVEGRP3)

____ DEFINE MAPSET(EVEH101) GROUP(EVEGRP3)

____ DEFINE MAPSET(EVEH130) GROUP(EVEGRP3)

____ DEFINE MAPSET(EVEH131) GROUP(EVEGRP3)

____ DEFINE MAPSET(EVEH135) GROUP(EVEGRP3)

____ DEFINE MAPSET(EVEH136) GROUP(EVEGRP3)

____ DEFINE MAPSET(EVEH137) GROUP(EVEGRP3)

____ DEFINE MAPSET(EVEH140) GROUP(EVEGRP3)

____ DEFINE MAPSET(EVEH150) GROUP(EVEGRP3)

____ DEFINE MAPSET(EVEH151) GROUP(EVEGRP3)

____ DEFINE MAPSET(EVEH152) GROUP(EVEGRP3)

____ DEFINE MAPSET(EVEH153) GROUP(EVEGRP3)

```

**If this is CICS/ESA 4.1, add the following definitions:**

```
*
* TRANSACTIONS for EVEGRP1 - Base and PPI
*

____ DEFINE TRANSACTION(COHO) PROGRAM(EVESECHO) GROUP(EVEGRP1)
 TASKDATALOC(ANY) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO PPI transaction)

____ DEFINE TRANSACTION(COPC) PROGRAM(EVESPPIC) GROUP(EVEGRP1)
 TASKDATALOC(ANY) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO PPI transaction)

____ DEFINE TRANSACTION(COPP) PROGRAM(EVESPPIP) GROUP(EVEGRP1)
 TASKDATALOC(ANY) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO stop PPI transaction)

____ DEFINE TRANSACTION(COPS) PROGRAM(EVESPPIS) GROUP(EVEGRP1)
 TASKDATALOC(ANY) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO start PPI transaction)

____ DEFINE TRANSACTION(COMC) PROGRAM(EVESCOMC) TWASIZE(512)
 TASKDATALOC(BELOW) TASKDATAKEY(USER) SHUTDOWN(ENABLED)
 DESCRIPTION(CICS AO CEMT Interface Transaction) GROUP(EVEGRP1)

____ DEFINE TRANSACTION(COMT) PROGRAM(EVESCEMT) TWASIZE(512)
 TASKDATALOC(BELOW) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO CEMT Interface Transaction) GROUP(EVEGRP1)

____ DEFINE TRANSACTION(COHR) PROGRAM(EVESHLTH) GROUP(EVEGRP1)
 TASKDATALOC(ANY) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO Health Checking Interface)

*
* PROGRAMS for EVEGRP1 - Base and PPI
*

____ DEFINE PROGRAM(EVESTISP) LANGUAGE(ASSEMBLER)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO enable PPI) GROUP(EVEGRP1)

____ DEFINE PROGRAM(EVESTIEX) LANGUAGE(ASSEMBLER)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO enable exits) GROUP(EVEGRP1)

____ DEFINE PROGRAM(EVESECHO) LANGUAGE(ASSEMBLER)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO Console Echo program) GROUP(EVEGRP1)

____ DEFINE PROGRAM(EVESCCCI) LANGUAGE(ASSEMBLER)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO PPI Interface Program) GROUP(EVEGRP1)

____ DEFINE PROGRAM(EVESC41) LANGUAGE(ASSEMBLER) RELOAD(NO)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO PPI XMEOUT user exit) GROUP(EVEGRP1)

____ DEFINE PROGRAM(EVESPERR) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
 DATALOCATION(ANY) EXECKEY(CICS)

____ DEFINE PROGRAM(EVESPMMSG) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO message handling program)

____ DEFINE PROGRAM(EVESPPIP) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO stop PPI program)

____ DEFINE PROGRAM(EVESPPI3) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO start PPI program)
```

```

____ DEFINE PROGRAM(EVESPPIU) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO start tran userid program)

____ DEFINE PROGRAM(EVESCEMT) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
 DATALOCATION(BELOW) EXECKEY(CICS)
 DESCRIPTION(CICS AO CEMT interface program)

____ DEFINE PROGRAM(EVESCOMC) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
 DATALOCATION(BELOW) EXECKEY(CICS)
 DESCRIPTION(CICS AO CEMT interface program)

____ DEFINE PROGRAM(EVESPINM) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO PPI function table)

____ DEFINE PROGRAM(EVESPLTT) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO PLTSD program)

____ DEFINE PROGRAM(EVESPPIC) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO PPI interface program)

____ DEFINE PROGRAM(EVESPPIS) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO start PPI program)

____ DEFINE PROGRAM(EVESHLTH) LANGUAGE(ASSEMBLER) GROUP(EVEGRP1)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO Health Checking Interface)

*
* TRANSACTIONS FOR EVEGRP2 - LMT
*

____ DEFINE TRANSACTION(COLO) PROGRAM(EVESYLOI)
 TASKDATALOC(ANY) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO LMT Operator Interface) GROUP(EVEGRP2)

____ DEFINE TRANSACTION(COLR) PROGRAM(EVESYPPS)
 TASKDATALOC(ANY) TASKDATAKEY(USER) GROUP(EVEGRP2)

____ DEFINE TRANSACTION(COLC) PROGRAM(EVESYLM I)
 TASKDATALOC(BELOW) TASKDATAKEY(USER) GROUP(EVEGRP2)

____ DEFINE TRANSACTION(COLE) PROGRAM(EVESYLM E)
 TASKDATALOC(ANY) TASKDATAKEY(USER) GROUP(EVEGRP2)

*
* programs for EVEGRP2 - LMT
*

____ DEFINE PROGRAM(EVESYPPI) LANGUAGE(ASSEMBLER)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT PPI program) GROUP(EVEGRP2)

____ DEFINE PROGRAM(EVESYPPS) LANGUAGE(ASSEMBLER)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT PPI program) GROUP(EVEGRP2)

____ DEFINE PROGRAM(EVESYLRQ) LANGUAGE(ASSEMBLER)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT requestor) GROUP(EVEGRP2)

____ DEFINE PROGRAM(EVESYLOI) LANGUAGE(ASSEMBLER)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT requestor) GROUP(EVEGRP2)

____ DEFINE PROGRAM(EVESYLS D) LANGUAGE(ASSEMBLER)
 DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)

```



```

 DESCRIPTION(CICS AO LMT main menu)

____ DEFINE PROGRAM(EVESYLDT) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT Operator interface)

____ DEFINE PROGRAM(EVESYLUS) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT Operator interface)

____ DEFINE PROGRAM(EVESYLSP) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program)

____ DEFINE PROGRAM(EVESYSPS) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program)

____ DEFINE PROGRAM(EVESYSPO) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program)

____ DEFINE PROGRAM(EVESYHLP) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT help)

____ DEFINE PROGRAM(EVESYLM) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
 DATALOCATION(BELOW) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program)

____ DEFINE PROGRAM(EVESYLMA) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program)

____ DEFINE PROGRAM(EVESYLME) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program)

____ DEFINE PROGRAM(EVESYLM)I LANGUAGE(ASSEMBLER) RELOAD(NO)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program)
 GROUP(EVEGRP2)

____ DEFINE PROGRAM(EVESYLMO) LANGUAGE(ASSEMBLER)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program)
 GROUP(EVEGRP2)

____ DEFINE PROGRAM(EVESYLMQ) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program)

____ DEFINE PROGRAM(EVESYLMR) LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program)

____ DEFINE PROGRAM(EVESYLM)S LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program)

____ DEFINE PROGRAM(EVESYLM)T LANGUAGE(ASSEMBLER) GROUP(EVEGRP2)
 DATALOCATION(BELOW) EXECKEY(CICS)
 DESCRIPTION(CICS AO LMT program)

*
* MAPSETS for LMT EUI - EVEGRP3
*

____ DEFINE MAPSET(EVEM100)
 GROUP(EVEGRP3)

____ DEFINE MAPSET(EVEM130)
 GROUP(EVEGRP3)

____ DEFINE MAPSET(EVEM135)
 GROUP(EVEGRP3)

```

|       |                        |                |
|-------|------------------------|----------------|
| _____ | DEFINE MAPSET(EVEM136) | GROUP(EVEGRP3) |
| _____ | DEFINE MAPSET(EVEM140) | GROUP(EVEGRP3) |
| _____ | DEFINE MAPSET(EVEM150) | GROUP(EVEGRP3) |
| _____ | DEFINE MAPSET(EVEM151) | GROUP(EVEGRP3) |
| _____ | DEFINE MAPSET(EVEM152) | GROUP(EVEGRP3) |
| _____ | DEFINE MAPSET(EVEM153) | GROUP(EVEGRP3) |
| _____ | DEFINE MAPSET(EVEH100) | GROUP(EVEGRP3) |
| _____ | DEFINE MAPSET(EVEH101) | GROUP(EVEGRP3) |
| _____ | DEFINE MAPSET(EVEH130) | GROUP(EVEGRP3) |
| _____ | DEFINE MAPSET(EVEH131) | GROUP(EVEGRP3) |
| _____ | DEFINE MAPSET(EVEH135) | GROUP(EVEGRP3) |
| _____ | DEFINE MAPSET(EVEH136) | GROUP(EVEGRP3) |
| _____ | DEFINE MAPSET(EVEH137) | GROUP(EVEGRP3) |
| _____ | DEFINE MAPSET(EVEH140) | GROUP(EVEGRP3) |
| _____ | DEFINE MAPSET(EVEH150) | GROUP(EVEGRP3) |
| _____ | DEFINE MAPSET(EVEH151) | GROUP(EVEGRP3) |
| _____ | DEFINE MAPSET(EVEH152) | GROUP(EVEGRP3) |
| _____ | DEFINE MAPSET(EVEH153) | GROUP(EVEGRP3) |

**If this is CICS TS R1, add the following definitions:**

```

*
* TRANSACTIONS for EVEGRP1 - Base and PPI
*

____ DEFINE TRANSACTION(COHO) PROGRAM(EVESECHO) GROUP(EVEGRP1)
 TASKDATALOC(ANY) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO PPI transaction)

____ DEFINE TRANSACTION(COPC) PROGRAM(EVESPPIC) GROUP(EVEGRP1)
 TASKDATALOC(ANY) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO PPI transaction)

____ DEFINE TRANSACTION(COPP) PROGRAM(EVESPPIP) GROUP(EVEGRP1)
 TASKDATALOC(ANY) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO stop PPI transaction)

____ DEFINE TRANSACTION(COPS) PROGRAM(EVESPPIS) GROUP(EVEGRP1)
 TASKDATALOC(ANY) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO start PPI transaction)

____ DEFINE TRANSACTION(COMC) PROGRAM(EVESCOMC) TWASIZE(512)
 TASKDATALOC(BELOW) TASKDATAKEY(USER) SHUTDOWN(ENABLED)
 DESCRIPTION(CICS AO CEMT Interface Transaction)
 GROUP(EVEGRP1)

____ DEFINE TRANSACTION(COMT) PROGRAM(EVESCEMT) TWASIZE(512)
 TASKDATALOC(BELOW) TASKDATAKEY(USER) GROUP(EVEGRP1)
 DESCRIPTION(CICS AO CEMT Interface Transaction)

____ DEFINE TRANSACTION(COHR) PROGRAM(EVESHLTH) GROUP(EVEGRP1)
 TASKDATALOC(ANY) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO Health Checking Interface)

____ DEFINE TRANSACTION(CORL) GROUP(EVEGRP1)
 DESCRIPTION(CICS AO RLS Status interface)
 PROGRAM(EVESRLSI) TWASIZE(0) PROFILE(DFHICIST)
 STATUS(ENABLED) TASKDATALOC(BELOW) TASKDATAKEY(USER)
 STORAGECLEAR(NO) RUNAWAY(SYSTEM) SHUTDOWN(DISABLED)
 ISOLATE(NO) DYNAMIC(NO) PRIORITY(1)
 TRANCLASS(DFHTCL00) DTIMOUT(NO) RESTART(NO)
 SPURGE(NO) TPURGE(NO) DUMP(YES) TRACE(YES)
 CONFDATA(NO) ACTION(BACKOUT) WAIT(YES)
 WAITTIME(0,0,0) RESSEC(NO) CMDSEC(NO)

*
* PROGRAMS for EVEGRP1 - Base and PPI
*

____ DEFINE PROGRAM(EVESRLSI) GROUP(EVEGRP1)
 DESCRIPTION(CICS AO RLS Status interface)
 LANGUAGE(ASSEMBLER) RELOAD(NO) RESIDENT(NO)
 USAGE(NORMAL) USELPACOPY(NO) STATUS(ENABLED)
 CEDF(YES) DATALOCATION(BELOW) EXECKEY(CICS)
 EXECUTIONSET(FULLAPI)

____ DEFINE PROGRAM(EVESTISP) LANGUAGE(ASSEMBLER)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO enable PPI) GROUP(EVEGRP1)

____ DEFINE PROGRAM(EVESTIEX) LANGUAGE(ASSEMBLER)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO enable exits) GROUP(EVEGRP1)

____ DEFINE PROGRAM(EVESECHO) LANGUAGE(ASSEMBLER)
 DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
 DESCRIPTION(CICS AO Console Echo program)

____ DEFINE PROGRAM(EVESCCCI) LANGUAGE(ASSEMBLER)
 DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
 DESCRIPTION(CICS AO PPI Interface Program)

```

```

| _____ DEFINE PROGRAM(EVESCM51) LANGUAGE(ASSEMBLER) RELOAD(NO)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 PPI XMEOUT user exit)
|
| _____ DEFINE PROGRAM(EVESPERR) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
|
| _____ DEFINE PROGRAM(EVESPMMSG) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 message handling program)
|
| _____ DEFINE PROGRAM(EVESPPIP) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 stop PPI program)
|
| _____ DEFINE PROGRAM(EVESPP13) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 start PPI program)
|
| _____ DEFINE PROGRAM(EVESPP1U) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 start tran userid program)
|
| _____ DEFINE PROGRAM(EVESCEMT) LANGUAGE(ASSEMBLER)
| DATALOCATION(BELOW) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 CEMT interface program)
|
| _____ DEFINE PROGRAM(EVESCOMC) LANGUAGE(ASSEMBLER)
| DATALOCATION(BELOW) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 CEMT interface program)
|
| _____ DEFINE PROGRAM(EVESPINM) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 PPI function table)
|
| _____ DEFINE PROGRAM(EVESPLTT) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 PLTSD program)
|
| _____ DEFINE PROGRAM(EVESPPIC) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 PPI interface program)
|
| _____ DEFINE PROGRAM(EVESPPIS) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 start PPI program)
|
| _____ DEFINE PROGRAM(EVESHLTH) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 Health Checking Interface)
|
| *
| * TRANSACTIONS FOR EVEGRP2 - LMT
| *
|
| _____ DEFINE TRANSACTION(COLO) PROGRAM(EVESYLOI)
| TASKDATALOC(ANY) TASKDATAKEY(USER) GROUP(EVEGRP2)
| DESCRIPTION(CICS A0 LMT Operator Interface)
|
| _____ DEFINE TRANSACTION(COLR) PROGRAM(EVESYPPS) GROUP(EVEGRP2)
| TASKDATALOC(ANY) TASKDATAKEY(USER)
|
| _____ DEFINE TRANSACTION(COLC) PROGRAM(EVESYLM1) GROUP(EVEGRP2)
| TASKDATALOC(BELOW) TASKDATAKEY(USER)
|
| _____ DEFINE TRANSACTION(COLE) PROGRAM(EVESYLM2) GROUP(EVEGRP2)
| TASKDATALOC(ANY) TASKDATAKEY(USER)
|
| *
| * programs for EVEGRP2 - LMT
| *

```

```

| _____ DEFINE PROGRAM(EVESYPPI) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT PPI program)
|
| _____ DEFINE PROGRAM(EVESYPPS) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT PPI program)
|
| _____ DEFINE PROGRAM(EVESYLRQ) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT requestor)
|
| _____ DEFINE PROGRAM(EVESYLOI) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT requestor)
|
|
| _____ DEFINE PROGRAM(EVESYLSL) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT main menu)
|
| _____ DEFINE PROGRAM(EVESYLDI) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT Operator interface)
|
| _____ DEFINE PROGRAM(EVESYLSU) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT Operator interface)
|
| _____ DEFINE PROGRAM(EVESYLSL) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT program)
|
| _____ DEFINE PROGRAM(EVESYSPS) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT program)
|
| _____ DEFINE PROGRAM(EVESYSPO) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT program)
|
| _____ DEFINE PROGRAM(EVESYHLP) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT help)
|
| _____ DEFINE PROGRAM(EVESYLM) LANGUAGE(ASSEMBLER)
| DATALOCATION(BELOW) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT program)
|
| _____ DEFINE PROGRAM(EVESYLMA) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT program)
|
| _____ DEFINE PROGRAM(EVESYLME) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT program)
|
|
| _____ DEFINE PROGRAM(EVESYLM1) LANGUAGE(ASSEMBLER) RELOAD(NO)
| DATALOCATION(ANY) EXECKEY(CICS)
| DESCRIPTION(CICS AO LMT program) GROUP(EVEGRP2)
|
| _____ DEFINE PROGRAM(EVESYLM0) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS)
| DESCRIPTION(CICS AO LMT program) GROUP(EVEGRP2)
|
| _____ DEFINE PROGRAM(EVESYLMQ) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT program)
|
| _____ DEFINE PROGRAM(EVESYLMR) LANGUAGE(ASSEMBLER)

```

```

| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT program)

| ____ DEFINE PROGRAM(EVESYLS) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT program)

| ____ DEFINE PROGRAM(EVESYLS) LANGUAGE(ASSEMBLER)
| DATALOCATION(BELOW) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT program)

| *
| * MAPSETS for LMT EUI - EVEGRP3
| *

| ____ DEFINE MAPSET(EVEM100) GROUP(EVEGRP3)
| ____ DEFINE MAPSET(EVEM130) GROUP(EVEGRP3)
| ____ DEFINE MAPSET(EVEM135) GROUP(EVEGRP3)
| ____ DEFINE MAPSET(EVEM136) GROUP(EVEGRP3)
| ____ DEFINE MAPSET(EVEM140) GROUP(EVEGRP3)
| ____ DEFINE MAPSET(EVEM150) GROUP(EVEGRP3)
| ____ DEFINE MAPSET(EVEM151) GROUP(EVEGRP3)
| ____ DEFINE MAPSET(EVEM152) GROUP(EVEGRP3)
| ____ DEFINE MAPSET(EVEM153) GROUP(EVEGRP3)
| ____ DEFINE MAPSET(EVEH100) GROUP(EVEGRP3)
| ____ DEFINE MAPSET(EVEH101) GROUP(EVEGRP3)
| ____ DEFINE MAPSET(EVEH130) GROUP(EVEGRP3)
| ____ DEFINE MAPSET(EVEH131) GROUP(EVEGRP3)
| ____ DEFINE MAPSET(EVEH135) GROUP(EVEGRP3)
| ____ DEFINE MAPSET(EVEH136) GROUP(EVEGRP3)
| ____ DEFINE MAPSET(EVEH137) GROUP(EVEGRP3)
| ____ DEFINE MAPSET(EVEH140) GROUP(EVEGRP3)
| ____ DEFINE MAPSET(EVEH150) GROUP(EVEGRP3)
| ____ DEFINE MAPSET(EVEH151) GROUP(EVEGRP3)
| ____ DEFINE MAPSET(EVEH152) GROUP(EVEGRP3)
| ____ DEFINE MAPSET(EVEH153) GROUP(EVEGRP3)

```

**If this is CICS TS R2, add the following definitions:**

```

*
* TRANSACTIONS for EVEGRP1 - Base and PPI
*

____ DEFINE TRANSACTION(COHO) PROGRAM(EVESECHO) GROUP(EVEGRP1)
 TASKDATALOC(ANY) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO PPI transaction)

____ DEFINE TRANSACTION(COPC) PROGRAM(EVESPPIC) GROUP(EVEGRP1)
 TASKDATALOC(ANY) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO PPI transaction)

____ DEFINE TRANSACTION(COPP) PROGRAM(EVESPPIP) GROUP(EVEGRP1)
 TASKDATALOC(ANY) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO stop PPI transaction)

____ DEFINE TRANSACTION(COPS) PROGRAM(EVESPPIS) GROUP(EVEGRP1)
 TASKDATALOC(ANY) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO start PPI transaction)

____ DEFINE TRANSACTION(COMC) PROGRAM(EVESCOMC) TWASIZE(512)
 TASKDATALOC(BELOW) TASKDATAKEY(USER) SHUTDOWN(ENABLED)
 DESCRIPTION(CICS AO CEMT Interface Transaction)
 GROUP(EVEGRP1)

____ DEFINE TRANSACTION(COMT) PROGRAM(EVESCEMT) TWASIZE(512)
 TASKDATALOC(BELOW) TASKDATAKEY(USER) GROUP(EVEGRP1)
 DESCRIPTION(CICS AO CEMT Interface Transaction)

____ DEFINE TRANSACTION(COHR) PROGRAM(EVESHLTH) GROUP(EVEGRP1)
 TASKDATALOC(ANY) TASKDATAKEY(USER)
 DESCRIPTION(CICS AO Health Checking Interface)

____ DEFINE TRANSACTION(CORL) GROUP(EVEGRP1)
 DESCRIPTION(CICS AO RLS Status interface)
 PROGRAM(EVESRLSI) TWASIZE(0) PROFILE(DFHICIST)
 STATUS(ENABLED) TASKDATALOC(BELOW) TASKDATAKEY(USER)
 STORAGECLEAR(NO) RUNAWAY(SYSTEM) SHUTDOWN(DISABLED)
 ISOLATE(NO) DYNAMIC(NO) PRIORITY(1)
 TRANCLASS(DFHTCL00) DTIMOUT(NO) RESTART(NO)
 SPURGE(NO) TPURGE(NO) DUMP(YES) TRACE(YES)
 CONFDATA(NO) ACTION(BACKOUT) WAIT(YES)
 WAITTIME(0,0,0) RESSEC(NO) CMDSEC(NO)

*
* PROGRAMS for EVEGRP1 - Base and PPI
*

____ DEFINE PROGRAM(EVESRLSI) GROUP(EVEGRP1)
 DESCRIPTION(CICS AO RLS Status interface)
 LANGUAGE(ASSEMBLER) RELOAD(NO) RESIDENT(NO)
 USAGE(NORMAL) USELPACOPY(NO) STATUS(ENABLED)
 CEDF(YES) DATALOCATION(BELOW) EXECKEY(CICS)
 EXECUTIONSET(FULLAPI)

____ DEFINE PROGRAM(EVESTISP) LANGUAGE(ASSEMBLER)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO enable PPI) GROUP(EVEGRP1)

____ DEFINE PROGRAM(EVESTIEX) LANGUAGE(ASSEMBLER)
 DATALOCATION(ANY) EXECKEY(CICS)
 DESCRIPTION(CICS AO enable exits) GROUP(EVEGRP1)

____ DEFINE PROGRAM(EVESECHO) LANGUAGE(ASSEMBLER)
 DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
 DESCRIPTION(CICS AO Console Echo program)

____ DEFINE PROGRAM(EVESCCCI) LANGUAGE(ASSEMBLER)
 DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
 DESCRIPTION(CICS AO PPI Interface Program)

```

```

| _____ DEFINE PROGRAM(EVESCM52) LANGUAGE(ASSEMBLER) RELOAD(NO)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 PPI XMEOUT user exit)
|
| _____ DEFINE PROGRAM(EVESPERR) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
|
| _____ DEFINE PROGRAM(EVESPMMSG) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 message handling program)
|
| _____ DEFINE PROGRAM(EVESPPIP) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 stop PPI program)
|
| _____ DEFINE PROGRAM(EVESPP13) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 start PPI program)
|
| _____ DEFINE PROGRAM(EVESPP1U) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 start tran userid program)
|
| _____ DEFINE PROGRAM(EVESCEMT) LANGUAGE(ASSEMBLER)
| DATALOCATION(BELOW) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 CEMT interface program)
|
| _____ DEFINE PROGRAM(EVESCOMC) LANGUAGE(ASSEMBLER)
| DATALOCATION(BELOW) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 CEMT interface program)
|
| _____ DEFINE PROGRAM(EVESPINM) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 PPI function table)
|
| _____ DEFINE PROGRAM(EVESPLTT) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 PLTSD program)
|
| _____ DEFINE PROGRAM(EVESPPIC) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 PPI interface program)
|
| _____ DEFINE PROGRAM(EVESPPIS) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 start PPI program)
|
| _____ DEFINE PROGRAM(EVESHLTH) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP1)
| DESCRIPTION(CICS A0 Health Checking Interface)
|
| *
| * TRANSACTIONS FOR EVEGRP2 - LMT
| *
|
| _____ DEFINE TRANSACTION(COLO) PROGRAM(EVESYLOI)
| TASKDATALOC(ANY) TASKDATAKEY(USER) GROUP(EVEGRP2)
| DESCRIPTION(CICS A0 LMT Operator Interface)
|
| _____ DEFINE TRANSACTION(COLR) PROGRAM(EVESYPPS) GROUP(EVEGRP2)
| TASKDATALOC(ANY) TASKDATAKEY(USER)
|
| _____ DEFINE TRANSACTION(COLC) PROGRAM(EVESYLM1) GROUP(EVEGRP2)
| TASKDATALOC(BELOW) TASKDATAKEY(USER)
|
| _____ DEFINE TRANSACTION(COLE) PROGRAM(EVESYLM2) GROUP(EVEGRP2)
| TASKDATALOC(ANY) TASKDATAKEY(USER)
|
| *
| * programs for EVEGRP2 - LMT
| *

```



```

| _____ DEFINE PROGRAM(EVESYPPI) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT PPI program)
|
| _____ DEFINE PROGRAM(EVESYPPS) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT PPI program)
|
| _____ DEFINE PROGRAM(EVESYLRQ) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT requestor)
|
| _____ DEFINE PROGRAM(EVESYLOI) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT requestor)
|
|
| _____ DEFINE PROGRAM(EVESYLSL) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT main menu)
|
| _____ DEFINE PROGRAM(EVESYLDI) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT Operator interface)
|
| _____ DEFINE PROGRAM(EVESYLSU) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT Operator interface)
|
| _____ DEFINE PROGRAM(EVESYLSL) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT program)
|
| _____ DEFINE PROGRAM(EVESYSPS) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT program)
|
| _____ DEFINE PROGRAM(EVESYSPO) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT program)
|
| _____ DEFINE PROGRAM(EVESYHLP) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT help)
|
| _____ DEFINE PROGRAM(EVESYLM) LANGUAGE(ASSEMBLER)
| DATALOCATION(BELOW) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT program)
|
| _____ DEFINE PROGRAM(EVESYLMA) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT program)
|
| _____ DEFINE PROGRAM(EVESYLME) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT program)
|
|
| _____ DEFINE PROGRAM(EVESYLM1) LANGUAGE(ASSEMBLER) RELOAD(NO)
| DATALOCATION(ANY) EXECKEY(CICS)
| DESCRIPTION(CICS AO LMT program) GROUP(EVEGRP2)
|
| _____ DEFINE PROGRAM(EVESYLM0) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS)
| DESCRIPTION(CICS AO LMT program) GROUP(EVEGRP2)
|
| _____ DEFINE PROGRAM(EVESYLMQ) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT program)
|
| _____ DEFINE PROGRAM(EVESYLMR) LANGUAGE(ASSEMBLER)

```

```

| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT program)
|
| ____ DEFINE PROGRAM(EVESYLSM) LANGUAGE(ASSEMBLER)
| DATALOCATION(ANY) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT program)
|
| ____ DEFINE PROGRAM(EVESYLMT) LANGUAGE(ASSEMBLER)
| DATALOCATION(BELOW) EXECKEY(CICS) GROUP(EVEGRP2)
| DESCRIPTION(CICS AO LMT program)
|
| *
| * MAPSETS for LMT EUI - EVEGRP3
| *
|
| ____ DEFINE MAPSET(EVEM100) GROUP(EVEGRP3)
|
| ____ DEFINE MAPSET(EVEM130) GROUP(EVEGRP3)
|
| ____ DEFINE MAPSET(EVEM135) GROUP(EVEGRP3)
|
| ____ DEFINE MAPSET(EVEM136) GROUP(EVEGRP3)
|
| ____ DEFINE MAPSET(EVEM140) GROUP(EVEGRP3)
|
| ____ DEFINE MAPSET(EVEM150) GROUP(EVEGRP3)
|
| ____ DEFINE MAPSET(EVEM151) GROUP(EVEGRP3)
|
| ____ DEFINE MAPSET(EVEM152) GROUP(EVEGRP3)
|
| ____ DEFINE MAPSET(EVEM153) GROUP(EVEGRP3)
|
| ____ DEFINE MAPSET(EVEH100) GROUP(EVEGRP3)
|
| ____ DEFINE MAPSET(EVEH101) GROUP(EVEGRP3)
|
| ____ DEFINE MAPSET(EVEH130) GROUP(EVEGRP3)
|
| ____ DEFINE MAPSET(EVEH131) GROUP(EVEGRP3)
|
| ____ DEFINE MAPSET(EVEH135) GROUP(EVEGRP3)
|
| ____ DEFINE MAPSET(EVEH136) GROUP(EVEGRP3)
|
| ____ DEFINE MAPSET(EVEH137) GROUP(EVEGRP3)
|
| ____ DEFINE MAPSET(EVEH140) GROUP(EVEGRP3)
|
| ____ DEFINE MAPSET(EVEH150) GROUP(EVEGRP3)
|
| ____ DEFINE MAPSET(EVEH151) GROUP(EVEGRP3)
|
| ____ DEFINE MAPSET(EVEH152) GROUP(EVEGRP3)
|
| ____ DEFINE MAPSET(EVEH153) GROUP(EVEGRP3)

```

---

## Step 6: DFHRPL and the CICS Automation Library

Add the AOCCICS.V1R4M0.SEVEMOD2 data set library to the DFHRPL concatenation.

---

## Chapter 9. CICS Automation Definitions in NetView

This section describes the definitions that take place in NetView. Use the following table to track your completed work:

| <i>Table 17. CICS Automation Definitions in NetView Check List</i> |                                                                      |
|--------------------------------------------------------------------|----------------------------------------------------------------------|
| √                                                                  | Step                                                                 |
| _____                                                              | 1: Basic CICS automation common control file definitions             |
| _____                                                              | 2: Basic CICS subsystem control file definitions                     |
| _____                                                              | 3: CICS Automation security checking                                 |
| _____                                                              | 4: Extended CICS automation common control file definitions          |
| _____                                                              | 5: Extended CICS subsystem common control file definitions           |
| _____                                                              | 6: The program-to-program interface initialization member (EVENTASK) |
| _____                                                              | 7: The REXX environment table                                        |
| _____                                                              | 8: Customizing the Status Display Facility                           |

---

### Step 1: Basic CICS Automation Common Control File Definitions

For each NetView domain, set up the CICS Automation automation environment according to the following checklist:

- \_\_\_ 1. Verify that the system environment has been defined as described in the SA OS/390 base documentation.

**An important note!**

You should already have a working and tested SA OS/390 before beginning the customization of CICS Automation.

- \_\_\_ 2. Use the worksheet shown in “Automation Operator Information Sheets” on page 312 to code the control file entry described in “AUTOOPS—Automation Operators” on page 50.
- \_\_\_ 3. Add the exits statement to the control file entry described in “ENVIRON SETUP—CICS Initialization Support” on page 53.

---

## Step 2: Basic CICS Subsystem Control File Definitions

For each CICS subsystem, set up the CICS Automation automation specifications according to the following checklist:

- \_\_\_ 1. Code the control file entry as described in “ACORESTART—NetView Restart Commands” on page 47.
- \_\_\_ 2. Use the worksheet shown in “Basic CICS Subsystem Information Sheets” on page 309 to code the subsystem information as described in the SA OS/390 base documentation. (Refer also to “SUBSYSTEM—Subsystem Definition” on page 57.)
- \_\_\_ 3. Use the worksheet shown in “Basic CICS Subsystem Information Sheets” on page 309 to code the APPLID= and MAJNODE= keywords in the control file entry described in “CICSCNTL - CICS extensions for SUBSYSTEM entry” on page 66.
- \_\_\_ 4. Read through the remaining CICSCNTL keywords and determine whether the defaults can be used. If not, change them at this time.
- \_\_\_ 5. Code the automation flags by subsystem as described in “Automation Flags” on page 48.
- \_\_\_ 6. Code the control file entry described in “THRESHOLDS—Error Thresholds” on page 58.
- \_\_\_ 7. Code the control file entry described in “STARTUP—Startup Commands” on page 56.
- \_\_\_ 8. Code the control file entry described in “Shutdown Control File Entries” on page 55.
- \_\_\_ 9. Add **SUBSYS=(subsys,AREA=CICS)**, where *subsys* is the symbolic name for this subsystem to the control file entry described in “PRODUCT - Subsystem State/Action Table sets” on page 77.

---

## Step 3: CICS Automation Security Checking

Edit EVESECUR and define, by class, those operators authorized to perform specific functions against specific CICS subsystems. The operator classes are coded KEYCLASS and VALCLASS. KEYCLASS can be a CICS subsystem name (as defined to CICS Automation), a domain, or a group name (also as defined to CICS Automation). VALCLASS identifies functions allowed for the subsystem, group, or domain specified with the previous KEYCLASS entry. All CICS subsystems must have security definitions in DSICMD.

Refer to the following example:

```
CICS1 KEYCLASS 1 <----- Operators with class 1 can work with CICS1.
INQUIRY VALCLASS 1 <----- Operators with class
1 can invoke an inquiry on CICS1.
STARTUP VALCLASS 2 <----- Operators with class
1 and 2 can invoke a startup on CICS1.
SHUTDOWN VALCLASS 2 <----- Operators with class
1 and 2 can invoke a shutdown on CICS1.
TRIGGERS VALCLASS 2 <----- Operators with class
1 and 2 can work with triggers on CICS1.
SERVICEP VALCLASS 2 <----- Operators with class
1 and 2 can work with service periods on CICS1.
CEMT VALCLASS 2 <----- Operators with class
1 and 2 can issue or set CEMTs on CICS1.
LMT VALCLASS 2 <----- Operators with class
1 and 2 can work with link monitoring on CICS1.
HEALTH VALCLASS 2 <----- Operators with class
1 and 2 can work with health checking on CICS1.
SUPPORT VALCLASS 2 <----- Operators with class
1 and 2 can work with support function on CICS1.
EVEPICT VALCLASS 2 <----- Operators with class
1 and 2 can start and stop the PPI on CICS1.
=OTHER VALCLASS 3 <----- Operators with class
1 and 3 can perform any other function on CICS1.
 not already defined in
this group of entries (default).

=OTHER KEYCLASS 9 <----- Operators with class
9 can work with any other CICS subsystem.

=OTHER VALCLASS 9 <----- Operators with class
9 can perform any function on any other
 CICS subsystem.
```

=OTHER is used as the default entry and can be used for either KEYCLASS or VALCLASS. There can only be a single =OTHER KEYCLASS specified, and it must be the last KEYCLASS entry.

**Note:** If you change any of the default KEYCLASS and VALCLASS values shipped in EVESECUR, then you must edit EVEPRFAO and add the new values to the OPCLASS statement. This will ensure that the AUTOTASKS which will be controlling various CICS regions and functions are authorized to do so. If this is not done, message EVE004I, plus EVE551E or EVE554E, may be received at STARTUP or SHUTDOWN. For example, if the AUTOTASK for a CICS is not authorized for health, then these messages will be received at shutdown:

- EVE004I SECURITY CHECK - OPID= CICSOP, KEYVALUE= HEALTH
- EVE551E UNEXPECTED RETURN CODE (RC=S) FROM EVEEMHCD IN EVEET020

You can use CICS Automation security checking for your own functions using the same framework described here. Security checking is called by the routine described in "CICSSEC - Invoke security checking" on page 96. The call syntax is:

**CICSSEC NAME=***subsys***,FUNCTION=***function*

where *function* is the function listed in DSICMD.

---

## Step 4: Extended CICS Automation Common Control File Definitions

If required, define the following control file entries:

- \_\_\_ 1. Use the worksheet shown in “CICS Subsystem Group Information Sheet” on page 315 to code the control file entry described in “CICSGROUP - Group of CICS subsystems” on page 69.

---

## Step 5: Extended CICS Subsystem Common Control File Definitions

If required, define the following control file entries:

- \_\_\_ 1. If you will use service periods, use the worksheet shown in “Service Periods Information Sheets” on page 317 to code the control file entry described in “SERVICE - Service period definitions” on page 83.
- \_\_\_ 2. If you will use link monitoring, use the worksheet shown in “Link Monitoring Information Sheets” on page 319 to code the control file entry described in “CONNECTION - Link monitoring definitions” on page 70.
- \_\_\_ 3. If you will use startup and shutdown triggers,
  - Use the worksheet shown in “Startup and Shutdown Trigger Information Sheets” on page 322 to code the control file entry described in “TRIGGER - Startup and shutdown triggers” on page 85.
  - Triggers can be unset when a startup is initiated, when a startup is complete, and when a shutdown is complete. Use the UNSET keyword described in “EXTCOND - External conditions” on page 73 to do this.
- \_\_\_ 4. If you will use the health check feature, use the worksheet shown in “Health Checking Information Sheets” on page 325 to define the HEALTHCHK keywords as described in “HEALTHCHK - Health checking” on page 75.
- \_\_\_ 5. If you will use customized State/Action Tables, code the control file entries described in “AREA - Define a set of State/Action Tables” on page 64 and “PRODUCT - Subsystem State/Action Table sets” on page 77.
- \_\_\_ 6. Review the control file sample (see “EVECFG01—Sample CICS Automation Control File Entries” on page 229) for entries not already described, such as:
  - LISTSHUT *subsys*
  - *subsys* PPIACTIVE,CMD='MVS...
  - *subsys* DFH1578,
  - *subsys* DFH1579,
  - *subsys* DFH1596,
  - RCVRSOS *subsys*.SOS
  - RCVRVIOL *subsys*.VIOL
  - ABCODETRAN *subsys*.TRAN
  - ABCODESYSTEM *subsys*.TRAN
  - and so on.

Customize these entries as required.

---

## Step 6: The program-to-program interface Initialization Member (EVENTASK)

This member only needs to be changed under the following circumstances:

- \_\_\_ 1. The BUFFQL is set to 20. If you get error messages indicating that 20 is not enough, increase this number. Refer to the *buffer queue limit* description in the *NetView Application Programming: Program-to-Program Interface* documentation for specific error messages and buffer queue guidelines.

Following is the default entry:

**BUFFQL=20**

- \_\_\_ 2. If you are running CICS Automation in more than one domain on the same MVS system, then you need to provide unique RECEIVERIDs in this member. This must also be changed in the corresponding CICS program-to-program interface initialization member EVESPINM. Following is the default entry:

**RECEIVERID=NETCVPP1**

- \_\_\_ 3. If you are using something other than the default program-to-program interface automation operator ID, as defined with the CICSPPI keyword described in “AUTOOPS—Automation Operators” on page 50, change that ID in the SERVER entries. The default is AUTCPPI, as shown in the following:

**SERVER=REQUEST,LMT,AUTCPPI,EVEEYPPS**  
**SERVER=RESPONSE,CEMT,AUTCPPI,EVESNRSP**  
**SERVER=RESPONSE,LMT,AUTCPPI,EVESNRSP**  
**SERVER=REQUEST,NACK,AUTCPPI,EVESNACK**  
**SERVER=RESPONSE,NACK,AUTCPPI,EVESNACK**

---

## Step 7: The REXX Environment Table

Refer to *AOC/MVS Planning and Installation* to review the number of REXX environments

---

## Step 8: Customizing the Status Display Facility

**Note:** Before using these steps, refer to the SA OS/390 base documentation for a complete overview on customizing the Status Display Facility. Also read “How to Set Up the Status Display Facility for CICS Automation” on page 14.

**Note:** The Status Display Facility members were merged in “Step 2: Copy CICS Automation Sample Members to the Target Library” on page 156 and “Step 3: Merge Status Display Facility Members” on page 156.

- \_\_\_ 1. Depending on how you are currently running your system, you have two ways of using the Status Display Facility panels. If you are running SUBSYS, where each individual system is not specifically defined in the tree, such as:

```

2 SYSTEM
 3 SUBSYS
2 CICS
 3 CICSHLTH
 3 VTAMACB
 3 CICSLMT
 3 CICSAUTO
 3 CICSMSG
 3 CICSSTOR
 4 CICSSES
 4 CICSVIOL
 3 CICSTRAN
 3 CICSTIMR

```

Then the supplied panels will all be used to display the information related to CICS.

If you are running without SUBSYS, such as:

```

2 SYSTEM
 3 JES
 3 VTAM
 3 CICS1
 3 CICS2

:
2 CICS
 3 CICSHLTH
 3 VTAMACB
 3 CICSLMT
 3 CICSAUTO
 3 CICSMSG
 3 CICSSTOR
 4 CICSSES
 4 CICSVIOL
 3 CICSTRAN
 3 CICSTIMR

```

with each CICS system specifically defined in the tree, then all the CICS entries except for CICSMSG will be placed on the system entry for that CICS when the alerts occur.

Depending on which method you choose, add an entry to your primary Status Display Facility panel definition member (the first panel displayed when either the SDF command is used) as shown:

**If you are not using SUBSYS**

```

SF(SY1.CICSMSG,05,74,75,N, ,EVEDMSG1)
ST(C)

```

**If you are using SUBSYS**

```

SF(SY1.CICS,05,74,75,N, ,EVED0001)
ST(C)

```

**Note:** These examples assume that your system name is SY1.



- \_\_\_ 2. Adjust the row, starting column, and ending column so that the letter C is adjacent to the system indicator (SY1), as shown in the following example:

| SYSTEM  |            | AOC/MVS SUPPORT SYSTEMS |          |       |           |          |     |
|---------|------------|-------------------------|----------|-------|-----------|----------|-----|
| System  | Subsystems | WTORs                   | Gateways | Spool | MVS Comps | Features |     |
| DALLAS  |            |                         |          |       |           | I        | C 0 |
| RALEIGH |            |                         |          |       |           |          | C   |
| CHICAGO |            |                         |          |       |           | I        |     |
| DENVER  |            |                         |          |       |           |          | 0   |
| PHOENIX | FPIMSEA    | IMS401E                 | A0F05I   | JES   |           | I        | C 0 |

- \_\_\_ 3. Add EVEPNLS to your existing AOFPNLS.
- \_\_\_ 4. If the system 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 Status Display Facility status field component name.

|              |              |              |              |              |
|--------------|--------------|--------------|--------------|--------------|
| ___ EVED0001 | ___ EVEDACB1 | ___ EVEDACB2 | ___ EVEDACBA | ___ EVEDACBB |
| ___ EVEDAUI1 | ___ EVEDAUI2 | ___ EVEDAUIA | ___ EVEDAUIB | ___ EVEDHLT1 |
| ___ EVEDHLT2 | ___ EVEDHLTA | ___ EVEDHLTB | ___ EVEDLMT1 | ___ EVEDLMT2 |
| ___ EVEDLMTA | ___ EVEDLMTB | ___ EVEDMSG1 | ___ EVEDMSG2 | ___ EVEDMSGA |
| ___ EVEDMSGB | ___ EVEDSTG1 | ___ EVEDSTG2 | ___ EVEDSTGA | ___ EVEDSTGA |
| ___ EVEDTIM1 | ___ EVEDTIM2 | ___ EVEDTIMA | ___ EVEDTIMB | ___ EVEDTRN1 |
| ___ EVEDTRN2 | ___ EVEDTRNA | ___ EVEDTRNB |              |              |

- \_\_\_ 5. Customize the panels as required.
- \_\_\_ 6. Code the CICS support definitions as described in “SDF—Status Display Facility Color Definitions” on page 52.

---

## Step 9: Adding a New CICS Automation Status File Record

If you define a new CICS region to CICS Automation and attempt to start using the operator interface, the startup will fail because there will be no status record for CICS Automation to read. To overcome this problem, you must build a status file record for each new CICS region, or you must trap a message for that region prior to attempting the manual startup. Any message trapped by CICS Automation for a CICS subsystem (such as IEF403I or DFH1500), will result in a status file record being built for that subsystem if one does not already exist. CICS Automation will dynamically build a status file record for each CICS subsystem upon trapping the first message for it.

To create a new status file record, enter:

```
EVESTS,ID=CICS_SUBSYSTEM_NAME,REQ=REPL
```

---

## Step 10: Installing CICSplex SM REXX API

To manage CICSplex SM CMAS address spaces the CPSM REXX API is required. This can be installed in NetView by adding the CICSplex SM library SEYUAUTH before the library that contains module IRXFLOC. If there are no existing IRXFLOC modules the library can be placed at the end of the //STEPLIB concatenation.

Alternatively, module IRXFLOC can be customized according to the instructions in the section "Installing the REXX function package" in manual *CICSplex SM Setup*.

---

## Chapter 10. Extended CICS Definitions

This chapter describes the CICS Automation definitions that take place in CICS.

### A helpful note

These steps are performed on each CICS region. Additional check lists are provided in Appendix F, “Extra Check Lists” on page 331.

Use the following table to track your completed work:

| Table 18. Extended CICS Definitions Check List |                                                              |
|------------------------------------------------|--------------------------------------------------------------|
| √                                              | Step                                                         |
| _____                                          | 1: Health check program and transaction definitions          |
| _____                                          | 2: Program-to-program interface initialization member (CICS) |
| _____                                          | 3: Add or change the CICS transient data messages            |
| _____                                          | 4: Echoplex back end programs                                |
| _____                                          | 5: Security considerations                                   |

---

### Step 1: Health Check Program and Transaction Definitions

**Note:** This step assumes that the health check routines are already written and that you know the transaction name, program name, and language of each. If you need more information, read “How to Use Health Checking” on page 28 before proceeding with this step.

If you will use the health check feature, do the following:

1. For each health check routine (there can be up to ten for each CICS subsystem) define the program name and language as follows:

DEFINE PROGRAM(*program*) LANGUAGE(*language*)

**Note:** The program name corresponds to the program name defined with the control file entry described in “HEALTHCHK - Health checking” on page 75.

---

### Step 2: Program-to-program interface Initialization Member (CICS)

**Note:** This step is only required if the initialization member requires modification. Modification is needed when:

1. The RECEIVERID is changed. This value must be the same as the value defined in “Step 6: The program-to-program interface Initialization Member (EVENTASK)” on page 187.
2. You are using the program-to-program interface for your own transactions.

The purpose of this initialization member is to relate program-to-program interface function names to CICS transaction names. The member looks something like this:

|                                                                           |                                                                                              |
|---------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| EVEMPINM TYPE=INITIAL,<br>BUFFQL=4,<br>RECEIVERID=NETVCPPI,<br>USERID=YES | INITIAL ENTRY<br>BUFFER QUEUE LIMIT<br>NPDS RECEIVER IDENTIFICATION<br>USE CICS 4.1 SECURITY |
| EVEMPINM TYPE=ENTRY,<br>FUNCTION=LMT,<br>TRANSID=COLR                     | DEFINE A FUNCTION<br>FUNCTION NAME<br>TRANSACTION NAME                                       |
| EVEMPINM TYPE=ENTRY,<br>FUNCTION=CEMT,<br>TRANSID=COMT                    | DEFINE A FUNCTION<br>FUNCTION NAME<br>TRANSACTION NAME                                       |
| EVEMPINM TYPE=ENTRY,<br>FUNCTION=HEALTH,<br>TRANSID=COHR                  | DEFINE A FUNCTION<br>FUNCTION NAME<br>TRANSACTION NAME                                       |
| EVEMPINM TYPE=FINAL                                                       | REQUIRED END                                                                                 |

If this member requires changes, do the following:

- \_\_\_ 1. Edit EVESPINM and change the USERID if required. Only CICS Version 4.1 and later systems will use this value. It will be ignored for all earlier versions of CICS.
- \_\_\_ 2. Use the JCL shown in “EVESJ020—Assemble and Link Edit Definition Members” on page 224 to assemble the program-to-program interface initialization member.  
  
**Note:** This JCL is used for assembling EVESCMT2 and EVESCMT3 as well. Edit the JCL and make sure that the member name to be assembled is EVEMPINM.
- \_\_\_ 3. Place the assembled member into one of the libraries in the CICSDFHRPL chain.

---

## Step 3: Add or Change the CICS Transient Data Messages

If you want to add or change the CICS transient data messages:

- \_\_\_ 1. Edit EVESCMT2 (for CICS/MVS) or EVESCMT3 (for CICS/ESA) and make the required changes.
- \_\_\_ 2. Use the JCL shown in “EVESJ020—Assemble and Link Edit Definition Members” on page 224 to assemble the table.  
  
**Note:** This JCL is used for assembling EVEMPINM as well. Edit the JCL and make sure that the member name to be assembled is EVESCMT2 or EVESCMT3.
- \_\_\_ 3. Place the assembled member into one of the libraries in the CICSDFHRPL chain.

---

## Step 4: Echoplex Back End Programs

If echoplexing is to be implemented, install the back end echoplex program to the remote systems according to the information in this table:

| If TYPE= | Use these definitions:                                                                                                                                                                                                                                                                                                                                |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CICS     | <pre>DEFINE TRANSACTION(ECHO) PROGRAM(EVESYCB7) DEFINE PROGRAM(EVESYCB7) LANGUAGE(ASSEMBLER)</pre> <p><b>Note:</b> The transaction name ECHO may be changed for your installation. Whatever name is used, it must be specified in the ECHO= keyword in the control file entry described in "CONNECTION - Link monitoring definitions" on page 70.</p> |
| IMS      | No installation required.                                                                                                                                                                                                                                                                                                                             |

---

## Step 5: Security Considerations

Perform this step only if you have CICS Version 4 Release 1 or higher.

If you wish to use non-terminal transaction security. Refer to the V4R1 *CICS Release Guide* for CICS specific information about non-terminal transaction security and the security manager domain.

- \_\_\_ 1. Define all NetView operators which will invoke CICS functions defined to RACF (or your SAF compliant security system). This will include:
  - Regular NetView operators
  - NetView autotasks which perform CICS related actions. These autotasks will include those autotasks specifically defined for CICS Automation use and may include the autotasks which process shutdown functions or resynchronization functions.
- \_\_\_ 2. Define SAF surrogate authorization for the CICS. Since CICS Automation is started from the PLTPI, surrogate authorization for the PLTPIUSR is recommended for all CICS Automation NetView users. Define surrogate profile types of DFHINSTL for PLTPI processing and DFHSTART for normal started task processing.
- \_\_\_ 3. Define TCICSTRN profiles for all the transactions which will use non-terminal transaction security. See "Step 2: Program List Table Definitions" on page 160 (in Chapter 8, Define CICS Automation to CICS) for the list of supplied CICS Automation transactions.
- \_\_\_ 4. Connect the NetView operators to the CICS resources in which they will need to access, such as transactions, programs and files. This connection is done through your SAF Security manager (such as RACF).
- \_\_\_ 5. Enable non-terminal transaction security in CICS by modifying the CICS SIT to specify XTRAN=YES and XUSER=YES.
- \_\_\_ 6. Modify PLTPIUSR and PLTPISEC as required.

If you do not plan on exploiting the non-terminal transaction security in CICS, disable this function in the AOC CICS feature by modifying the EVESPINM member and specifying USERID=NO to disable extended support. Reassemble the member using sample job EVESJ020.



---

## Chapter 11. CICS Automation Definitions for CICSplex System Manager (CPSM)

This chapter is a guide to help you define CICSplex System Manager components to Automation.

---

### Automating Coordinating Address Space (CAS) Startup and Shutdown

The CAS applications are not CICS systems. System Automation for OS/390 should be used to manage them. CAS start and stop procedures should be defined to the System Automation for OS/390.

To help with the definition of the CAS the CICS Feature supplies NetView Automation Table entries to let Automation know when the CAS is UP. These definitions are stored in member EVEMEYU1 in the Samples library.

To assist in the task of building the ACF policy, a sample subsystem definition for a CAS has been provided. Sample member EVECFG01 in SEVESAMP contains a subsystem definition for a CAS region named EYUCAS1A.

---

### Automating CICSplex SM Address Space (CMAS) Startup and Shutdown

The CMAS regions are CICS regions, and so the CICS Feature can be used to automate the startup and shutdown of these regions. The CMAS regions execute only CICSplex SM code. (CICSplex SM recommends that only CICSplex code be run in CMAS regions. User transactions should not be run in CMAS regions.)

The Automation required to manage the CMAS regions is less than a normal CICS region because there is no need to have PPI and other CICS monitoring functions. However, CMAS CICS regions need to be shutdown via the CICSplex SM SHUTDOWN command. To allow the CICS Feature to shutdown the CMAS, a new command, CMASSHUT, has been created. Use it in place of the normal CICSSHUT command in the ACF policy definition for CMAS regions.

**Note:** The use of CEMT PERFORM SHUTDOWN is not recommended for CMAS regions.

CPSM recommends that CMAS regions be started prior to MAS regions they manage. This can be achieved via the parent/child hierarchy as supported by System Automation for OS/390.

To assist in the task of building the ACF Policy, two Class definitions have been created. One for the automation of normal CICS Transaction Server for OS/390 regions and one for CMAS regions running on CICS Transaction Server for OS/390.

Member EVECFG01 in SEVESAMP contains the new class definitions. The SUBSYSTEM\_CLASS for normal CICS Transaction Server for OS/390 regions is CPSMCICS12. The SUBSYSTEM\_CLASS for CMAS regions running on CICS Transaction Server for OS/390 is CPSMCMAS12. To install these classes, uncomment the definition statements in EVECFG01. Alternatively, the sample class

| definitions can be extracted to a work member and the definitions uncommented.  
| Then these definitions can be migrated into the Dialogs via the MIGRATE function.  
| Once in the dialogs, they can be used as required.

| A sample SUBSYSTEM definition using these classes is provided in commented  
| form. For the CICS Transaction Server for OS/390 class this is EYUMAS1A. For  
| the CMAS this is EYUCMS1A.



---

## Part 4. Appendixes



## Appendix A. CICS Automation and the Program-to-Program Interface

### Warning!

CICS Automation uses the program-to-program interface for issuing CEMT transactions from the operator interface, for link monitoring, and for health checking. If you are considering using the CICS Automation program-to-program interface code for your own purposes, remember that this interface is release sensitive. This significance of this is that you may need recompile or make changes to your code to be compatible with new releases of CICS Automation or NetView.

## Program-to-Program Interface Components in NetView and CICS

Figure 11 illustrates CICS Automation program-to-program interface components in NetView and in CICS.

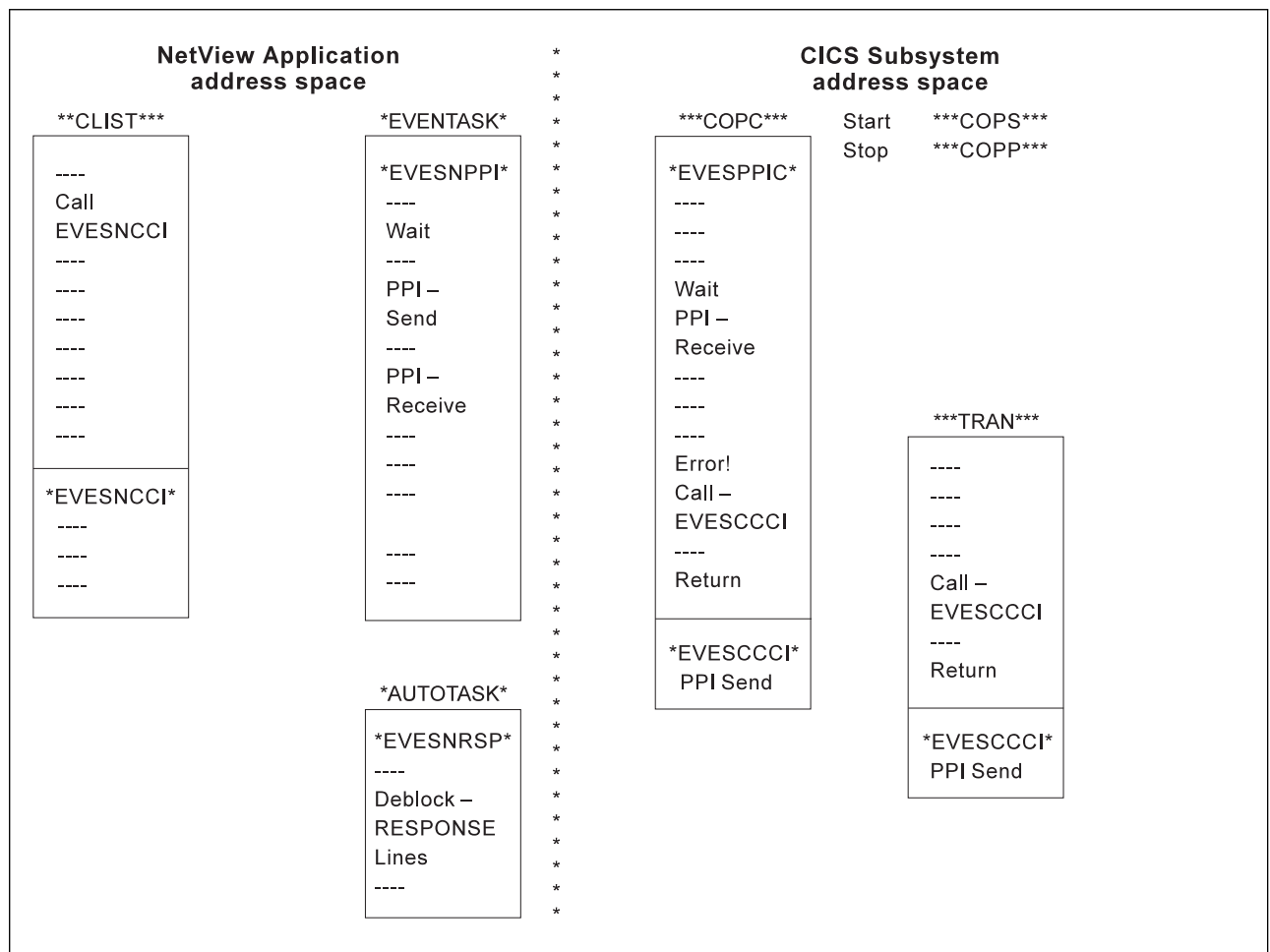


Figure 11. Program-to-program interface Components in NetView and CICS. Two of the programs shown in this figure have not been previously mentioned: EVESNPPI and EVESPPIC. EVESNPPI is the NetView subtask program. EVESPPIC is the CICS long running receiver program. These are internal programs and are not described in this manual.

---

## NetView Requests Using the Program-to-Program Interface

The following requests from NetView are described in this section:

- CONVERSE
- SEND
- CANCEL.

EVESNCCI is used for these requests. This section only provides an overview of how EVESNCCI works. The programming details, such as command syntax, return codes, and segment support, are provided in “EVESNCCI - NetView to CICS Communication Interface” on page 207.

### CONVERSE from NetView

A CONVERSE request from a NetView command list (or command processor) starts a CICS transaction on the same host. The CICS transaction is expected to return a response to a specified NetView task in a named NetView domain. The response can have the form of a RESPONSE, an ACK, or a NACK.

The EVESX001 (CICSSEC) and EVESX022 (CICSQRY) command processors are called to verify the authorization of the caller and to obtain the VTAM application identifier which is used as the program-to-program interface receiver identification. EVESNCCI returns a message and a return code indicating whether the command was successful or not. If the command is accepted, it is forwarded to the EVENTASK optional task.

The EVENTASK optional task uses the program-to-program interface to notify CICS that the transaction is to be started.

The started transaction retrieves the input data. The CICS transaction returns a response to NetView. The response is sent back to the EVENTASK optional task using the program-to-program interface. The processing of the response sent to NetView differs for the various response types, as described below:

#### A for ACK

The following message is sent to the requestor:

```
EVE128I POSITIVE ACKNOWLEDGEMENT
```

#### N for NACK

The following message is sent to the requestor:

```
EVE129E response-data
```

where *response-data* is the data is the data returned by the CICS program.

#### R for RESPONSE

When this request type is used, the NetView program-to-program interface initialization member is interrogated to locate the function requested so that the command list can be identified. The request is then scheduled under the autotask associated with the requested function, after which the data is sent to the requestor.

The following figure illustrates the flow of events initiated by an EVESNCCI CONVERSE request:

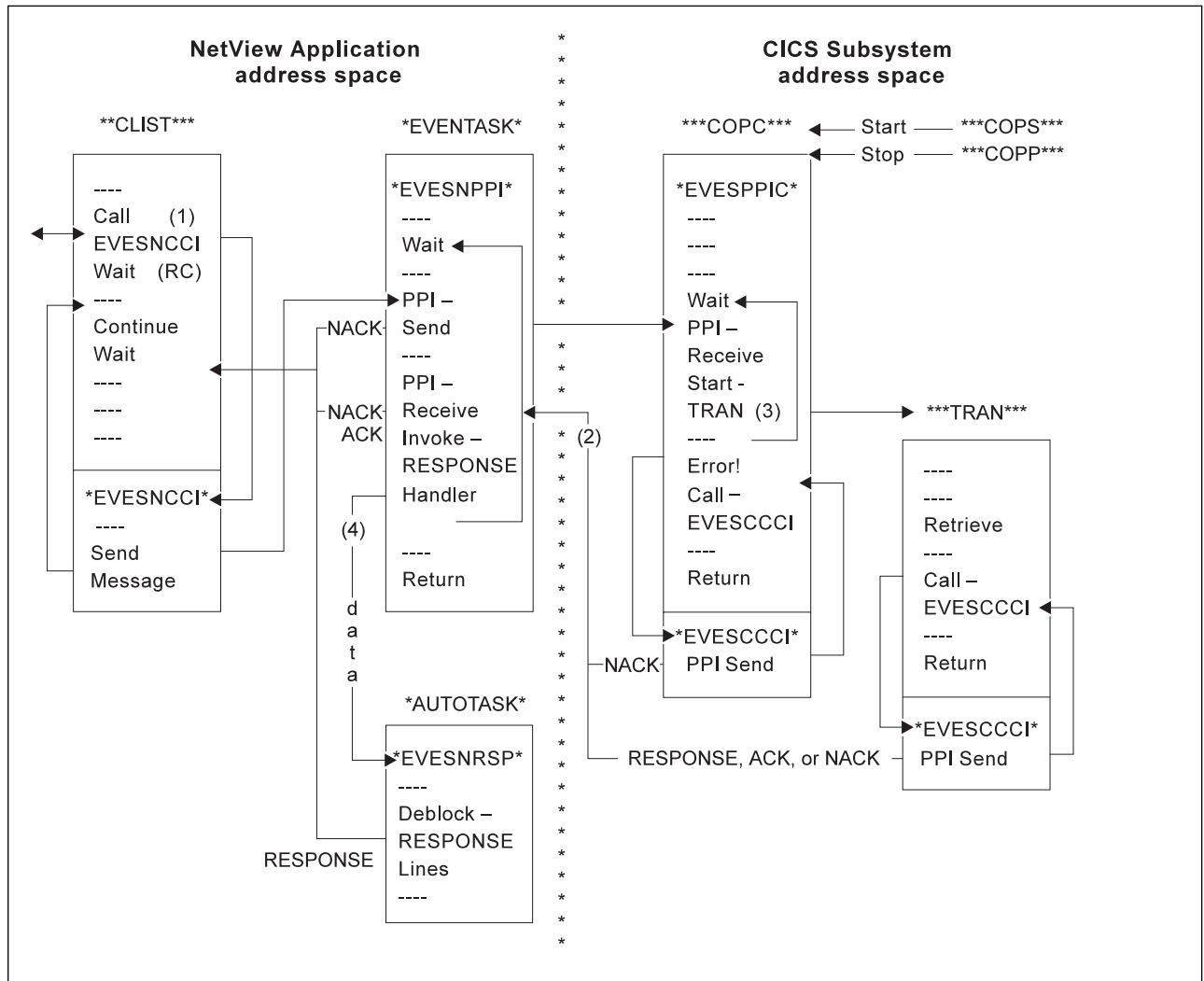


Figure 12. An EVESNCCI CONVERSE Request

#### Notes:

- Parameters passed with EVESNCCI indicate the request type (in this case C for CONVERSE), the target CICS, the name of the function to be invoked, data (if required), and the requesting operator ID and domain ID.
- The responses (RESPONSE, ACK, or NACK) are returned from CICS to the requestor (operator ID and domain ID).
- The function name is used to locate the transaction name in the CICS initialization member. If the transaction name cannot be found, a NACK response is returned.
- The name of the autotask and the name of the command processor or command list in NetView are obtained using the function name in the NetView initialization member. If the name or names cannot be found, or if the scheduling of the autotask fails, a NACK response is returned.

## SEND from NetView

A SEND request from a NetView command list starts a CICS transaction on the same host. No response is returned. The EVESX001 (CICSSEC) and EVESX022 (CICSQRY) command processors verify the authorization of the caller and obtain the VTAM application identifier which is used as the program-to-program interface receiver identification. EVESNCCI returns a message and return code indicating whether the command was successful or not. If the command is accepted, it is forwarded to the EVENTASK optional task. This task uses the program-to-program interface to notify CICS to start the transaction.

Errors found by EVESNCCI are returned to the caller. Other errors detected during the processing of a SEND request are not returned. These errors are only logged.

Refer to the following figure:

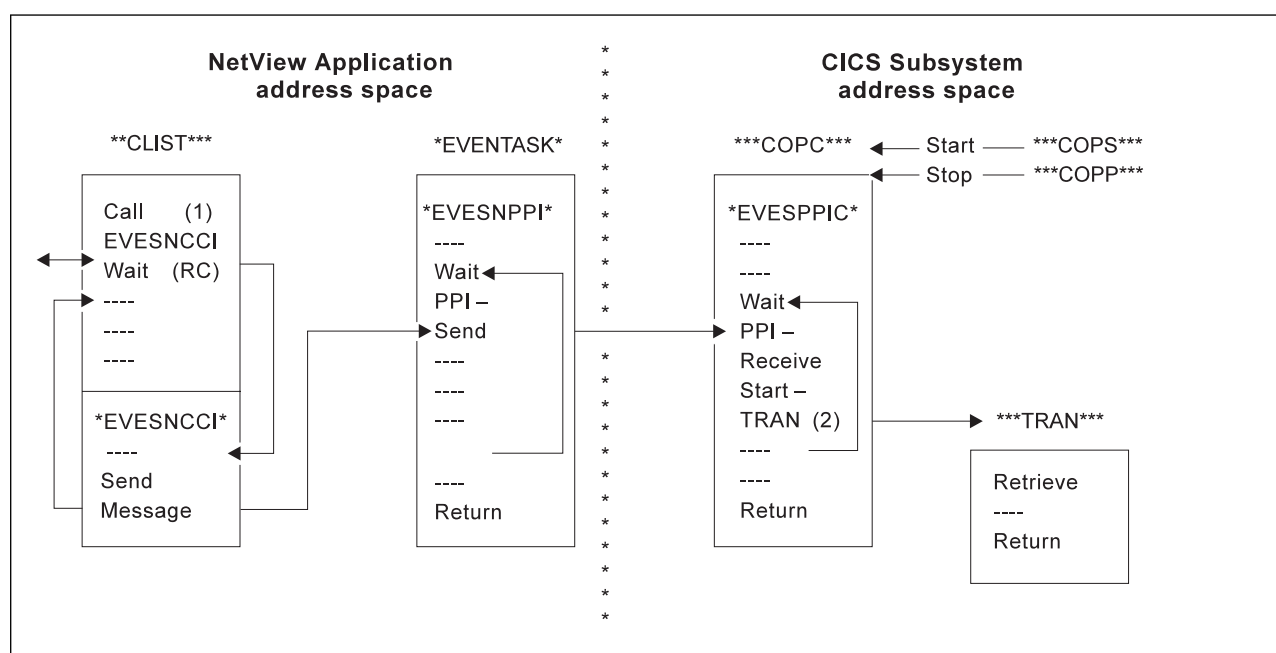


Figure 13. An EVESNCCI SEND Request

### Notes:

1. Parameters passed with EVESNCCI indicate the request type (in this case S for SEND), the target CICS, the name of the function to be invoked, and data (if required).
2. The function name is used to locate the transaction name in the CICS initialization member.

## CANCEL from NetView

The maximum amount of text that can be specified on the EVESNCCI command is limited to 240 bytes, including the command name EVESNCCI. The maximum amount of data that can be specified varies with the particular EVESNCCI command, but it is always less than 240 bytes. The CICS Automation program-to-program interface implementation provides segment support which allows up to 32656 bytes to be sent from NetView to another program-to-program interface receiver. Segment support is described in "EVESNCCI - NetView to CICS

Communication Interface” on page 207 (refer to the SEGMENT= keyword and the usage notes).

The EVESNCCI CANCEL request is used when the segment assembly process must be terminated. This request type causes all saved segments for the specified segment identifier to be freed. If the command is accepted, it is always successful, whether saved segments with the specified segment identifier exist or not.

Users of the segment function are requested to use CANCEL when applicable. This prevents the NetView application system to be filled with unused data.

Errors found by EVESNCCI are returned to the caller. Each CANCEL is logged.

---

## CICS Requests Using the Program-to-Program Interface

The following requests from CICS are described in this section:

- CONVERSE
- SEND.

EVESCCCI is used for these requests. This section only provides an overview of EVESCCCI. The programming details, such as command syntax and return codes, are provided in “EVESCCCI - CICS to NetView Communication Interface” on page 213.

### CONVERSE from CICS

A CONVERSE request from a CICS transaction starts a command list or a command processor in the NetView application system on the same host. The command list or command processor is expected to return a response to the CICS transaction. The response can have the form of a RESPONSE, an ACK, or a NACK.

If the request is accepted, EVESCCCI sends the request to the EVENTASK optional task. The EVENTASK optional task translates the function specification to a command list or command processor name using the EVENTASK initialization member. The autotask under which the command list or processor is to be executed is also obtained from the EVENTASK initialization member.

Errors found by EVESCCCI are returned to the caller. When other errors are detected during the processing of a CONVERSE request, CICS Automation uses the program-to-program interface to return a NACK response to the CICS transaction. The NACK response data indicates the type of error that occurred. The NACK response is logged via message EVE180E.

Refer to Figure 14 on page 204.

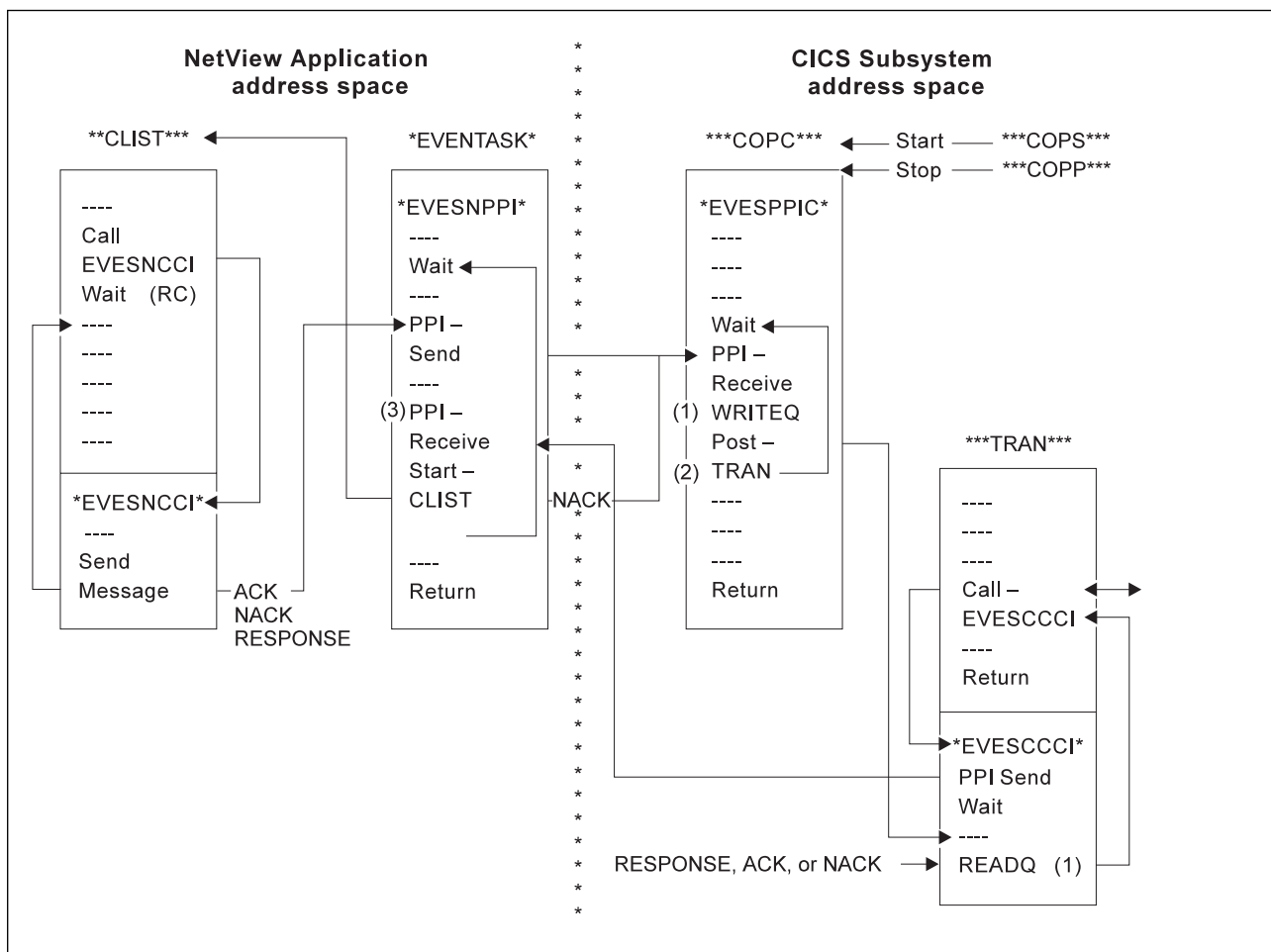


Figure 14. An EVESCCCI CONVERSE Request

#### Notes:

1. The received data is saved and obtained from temporary storage.
2. The Post-tran is actually a CANCEL of an interval control request.
3. The EVENTASK optional task translates the function specification to a command list or command processor name using the EVENTASK initialization member. The autotask under which the command list or processor is to be executed is also obtained from the EVENTASK initialization member. If the name cannot be obtained or if the scheduling of the command list fails, a NACK response is returned.

## SEND from CICS

A SEND request from a CICS transaction starts a command list or a command processor in NetView. No response is returned by the command list or command processor to the CICS transaction.

If the request is accepted, EVESCCCI sends the request to the EVENTASK optional task. The EVENTASK optional task translates the function specification to a command list or command processor name using the EVENTASK initialization member. The autotask under which the command list or processor is to be executed is also obtained from the EVENTASK initialization member. Then, the NetView command processor or command list returns to NetView.



Errors found by EVESCCCI are returned to the caller via a return code and an error code. Other errors detected during the processing of a SEND request are not returned to the requestor. The errors are only logged.

Refer to the following figure:

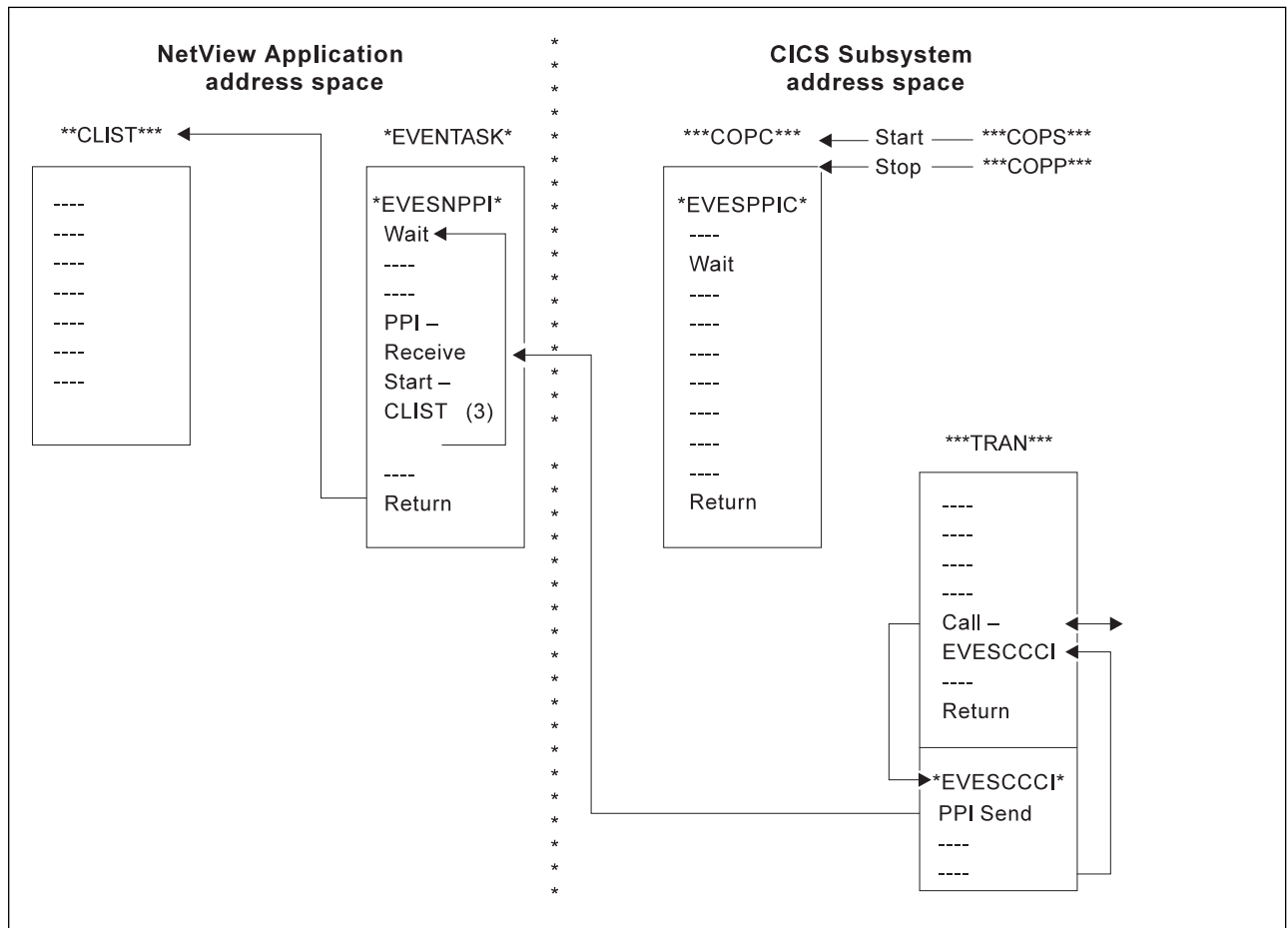


Figure 15. An EVESCCCI SEND Request

## Programming Interface

The CICS Automation CICS to NetView program-to-program interface members are:

### “EVESNCCI - NetView to CICS Communication Interface” on page 207

Allows you to:

- Initiate, from NetView, the execution of a CICS transaction.
- Send a response to a CICS transaction.

### “EVESNRSP - Common Response Handler from CICS” on page 212

The maximum length of a RESPONSE response line is 216 characters. EVESNRSP unblocks the response data in NetView and turns it into a multi-line WTO by calling the EVESX002 (CICSBMSG) command processor for each response line.

**“EVESCCCI - CICS to NetView Communication Interface” on page 213.**

Use this subroutine to request, from a CICS transaction, the initiation of a NetView command, a command list, or command processor in NetView.

**“EVEMPINT - EVESCCCI Parameter List Copy Book” on page 216.**

The EVEMPINT copy book describes the CICS Automation program-to-program interface parameter list at the CICS side.

---

## EVESNCCI - NetView to CICS Communication Interface

Use this subroutine to:

- Initiate, from NetView, the execution of a CICS transaction.
- Send a response to a CICS transaction.

```
EVESNCCI TYPE=C|S|R|A|N|X
 ,NAME=subsnm
 ,FUNC=function
 [,DATA=data]
 [,REQID=reqid]
 [,SEGMENT=segment]
 [,ID=id]
```

### Keyword and Parameter Definitions

#### TYPE=

Specifies the type of command. Valid command types are:

- C** For CONVERSE: Starts a process at the other end. A response is expected.
- S** For SEND: Starts a process at the other end. No response is expected.
- R** For RESPONSE: This is used to send data to the CICS transaction that issued the CONVERSE request.
- A** For ACK: Signals the successful completion of a CONVERSE request. No data is provided.
- N** For NACK: Signals the unsuccessful completion of a CONVERSE request. Data can optionally be provided (maximum of 100 bytes).
- X** For CANCEL: Use this request type to cancel the segment assembly process.

#### NAME=

Specifies the CICS subsystem name as it is known to CICS Automation.

#### FUNC=

Specifies the symbolic name of a process to be initiated or to be responded to, such as a CICS transaction or a NetView command list. The relation between a function name and a process name is contained in the EVENTASK and EVESPINM initialization members.

## EVESNCCI - NetView to CICS Communication Interface

### DATA=

Specifies the request or response data. Not accepted for ACK responses. The maximum data length on a NACK response is 100 bytes. If omitted, no data is passed.

Three data delimiter pairs are supported: single quotes, double quotes, or parentheses. These delimiters must be used when the data contains blanks or commas, or starts with one of the data delimiters itself. Data delimiters are stripped off before passing the data on.

### REQID=

Specifies the request identification. This field relates the response to the CICS transaction requesting the response. The value is provided on the CICS CONVERSE request and should simply be copied. The request identification may not contain commas or blanks. It is required and only accepted for responses.

### SEGMENT=

Use this keyword when the EVESNCCI command must be longer than 240 bytes. The SEGMENT= parameters are:

- F** First segment.
- M** Neither the first and nor the last segment.
- L** Last segment.

When the SEGMENT= keyword is used, the ID= keyword is used to identify the segment chain within a NetView task.

### ID=

Identifies the segment chain within a NetView task. This data field is 16 bytes. The segment chain identification may not start with DSI or EVE and may not contain commas or blanks.

## Comments and Usage Notes

1. The old parameters **DOMAIN=** and **OPID** are ignored.
2. The requested function must be defined in the CICS Automation program-to-program interface initialization members, as shown for the CEMT function:

#### In NetView (EVENTASK)

```
SERVER=RESPONSE,CEMT,AUTCPPI,EVESNRSP
```

#### In CICS (EVEMPINM)

|                      |                   |
|----------------------|-------------------|
| EVEMPINM TYPE=ENTRY, | DEFINE A FUNCTION |
| FUNCTION=CEMT,       | FUNCTION NAME     |
| TRANSID=COMT         | TRANSACTION NAME  |

3. The maximum amount of text that can be specified on the EVESNCCI command is limited to 240 bytes, including the command name EVESNCCI.
4. The maximum amount of text that can be specified with a segment chain is 32656 bytes.
5. EVESNCCI saves the segments until the final segment is received. After receiving the final segment, EVESNCCI assembles the segments into one

## EVESNCCI - NetView to CICS Communication Interface

block which is sent to the specified receiver using the EVENTASK optional task.

**Note:** It is assumed that the user of the segment function provides the segments in the correct order and that the segment identification is unique within the NetView task.

6. If an internal error is found during processing of a segment request, such as a GETMAIN failure, all existing segments are deleted.
7. All CANCEL commands (initiated when TYPE=X is used) are logged together with their results. The CANCEL command is always successful. Users of the segment function are requested to use TYPE=X when applicable to prevent NetView from being filled with unused data.
8. The following matrix shows the required (R), optional (O), and invalid (¬) keywords for the various command types.

| TYPE= | NAME | FUNC | DATA           | OPID | DOMA | REQUI | SEGM           | ID             |
|-------|------|------|----------------|------|------|-------|----------------|----------------|
| C     | R    | R    | O              | O    | O    | ¬     | O <sub>a</sub> | O <sub>d</sub> |
| S     | R    | R    | O              | ¬    | ¬    | ¬     | O <sub>a</sub> | O <sub>d</sub> |
| R     | R    | R    | O              | ¬    | ¬    | R     | O <sub>a</sub> | O <sub>d</sub> |
| N     | R    | R    | O <sub>b</sub> | ¬    | ¬    | R     | ¬              | ¬              |
| A     | R    | R    | ¬              | ¬    | ¬    | R     | ¬              | ¬              |
| X     | ¬    | ¬    | ¬              | ¬    | ¬    | ¬     | ¬              | R              |
| none  | ¬    | ¬    | O              | ¬    | ¬    | ¬     | R <sub>c</sub> | R <sub>d</sub> |

### Notes:

- a. SEGMENT=F only can be used with these TYPEs.
  - b. The length of the data must be less than 101 bytes.
  - c. SEGMENT=M or SEGMENT=L can only be used when no TYPE is specified.
  - d. When a SEGMENT is specified, an ID must also be given.
9. EVESNCCI returns the following return codes and error messages to the caller. Some of the messages are written to the NetView log. Message EVE122E is also sent to the authorized receiver.

**RC=0** EVE120I COMMAND ACCEPTED FOR *subsys*, APPLID = *applid*

The command has been forwarded to the EVENTASK optional task where *subsys* is the symbolic name by which this CICS subsystem is known to CICS Automation and *applid* is the generic VTAM application identifier of the target CICS subsystem.

**RC=4** EVE121E ERROR ON DSIxxx REQUEST IN EVESNCCI, RC=ccc

The command has not been forwarded to the EVENTASK optional task where *xxx* identifies the DSI request (such as GET, FRE, FIND, PUSH, or POP) and *ccc* is the return code returned by the DSIxxx function.

**RC=8** EVE122E EVENTASK TASK NOT ACTIVE

The command has not been forwarded to the EVENTASK optional task.

### **RC=12** EVE123E INPUT ERROR AT DISPLACEMENT *ddd*, CODE=*ccc*

The command has not been forwarded to the EVENTASK optional task where *ddd* contains the location in the command string where the error was detected and *ccc* is one of the following:

- 004** Unrecognized keyword.
- 008** Syntax error.
- 012** Operand error.
- 016** Duplicate keyword.
- 020** Conflicting keyword.
- 024** Required keyword(s) omitted.
- 028** Incorrect data length. The data length on an ACK response was not zero or the data length on a NACK response was larger than 100 bytes.

### **RC=16** EVE124E SEGMENT ERROR, CODE = *ccc*

The command has not been forwarded to the EVENTASK optional task. All existing segments that have the current ID are deleted, except when the segment-chain was corrupted, where *ccc* is one of the following:

- 004** SEGMENT SEQUENCE ERROR. A middle or last segment has been offered while no first segment with identical ID was available, or a first segment has been offered while another first segment with the same ID already exists.
- 008** TOO MUCH DATA. In a series of segments with identical ID the total amount of data exceeds 32656 bytes.
- 012** SEGMENT-CHAIN CORRUPTED. Storage used for saving segment data has been overwritten.

### **RC=20** EVE125E NO STORAGE AVAILABLE ON DSIxxx REQUEST IN EVESNCCI

The command has not been forwarded to the EVENTASK optional task.

### **RC=24** ERROR ON EVESXnnn CALL IN EVESNCCI, RC = *ccc*

The command has not been forwarded to the EVENTASK optional task. EVESNCCI calls the EVESX001 (CICSSEC) and EVESX022 (CICSQRY) command processors. A non-zero return code from either of these command processors results in this return code and error message. This error implies (normally) that either the caller is not authorized for the function on the specified subsystem or the specified subsystem is not defined.

10. When other errors are detected during the processing of a CONVERSE request, CICS Automation program-to-program interface returns a NACK response to the requestor. The NACK response data indicates the type of error that occurred. The errors are also logged.

The following NACK responses can be expected:

```
EVE129E text
 EVE122E task TASK NOT ACTIVE
 EVE136E ERROR ON PPI REQUEST rrr, RC = ccc
 EVE137E NETVIEW SUBSYSTEM NOT AVAILABLE
 EVE141E INCORRECT MQS BUFFER RECEIVED IN progrname
 EVE142E FUNCTION funcname NOT FOUND IN membrname
 EVE171E procname : ERROR IN progrname(tran), CODE = cccc
 EVE175E procname : FUNCTION funcname NOT FOUND IN EVESPINM
 EVE181E procname : ERROR ON TRANSACTION START tran FOR FUNCTION funcname
```

### Examples of Usage

#### Example 1

```
"EVESNCCI TYPE=C,"||,
 "NAME=CICS1,"||,
 "FUNC=CEMT,"||,
 "DATA='I PR(E*)'"
```

This command starts a CEMT transaction in the CICS *subsystem* known to CICS Automation as CICS1.

#### Example 2

```
"EVESNCCI TYPE=R,"||,
 "NAME=CICS2,"||,
 "FUNC=LMT,"||,
 "DATA=(1st segment of LMT response),"||,
 "REQID=1234567890123456,"||,
 "SEGMENT=F,"||,
 "ID=QAZWSXEDCRFVTGBY"

"EVESNCCI SEGMENT=M,"||,
 "DATA=(2nd segment of LMT response),"||,
 "ID=QAZWSXEDCRFVTGBY"

"EVESNCCI SEGMENT=L,"||,
 "DATA=(3rd segment of LMT response),"||,
 "ID=QAZWSXEDCRFVTGBY"
```

This is a link monitor response, which is in 3 segments. Error logic is not included.

---

### EVESNRSP - Common Response Handler from CICS

The maximum length of a RESPONSE response line is 216 characters. EVESNRSP deblocks the response data in NetView and turns it into a multi-line WTO by calling the EVESX002 (CICSBMSG) command processor for each response line.

#### Comments and Usage Notes

1. The request data is scanned for the presence of NL (X'15') characters within the maximum response line length. If a NL character is found, it delimits the current response line. If no NL character is found, a maximum length response line is assumed, or the end of the response data delimits the current response line.
2. The following EVESX002 (CICSBMSG) commands are issued by EVESNRSP:  

```
EVESX002 START,"domainid,opid
EVESX002 DATA,C,EVE790I PPI RESPONSE FROM applid FOR FUNCTION function maxln totln
EVESX002 DATA,D,response line text
EVESX002 DATA,E,EVE792I END"
```

This results in the following multi-line WTO to be sent to *opid* on *domainid*:

```
EVE790I PPI RESPONSE FROM applid FOR FUNCTION function maxln totln
response line text
:
response line text
EVE792I END
```

where *maxln* and *totln* contain the maximum response line length and the total length of the RESPONSE response data sent to NetView.

3. The maximum value of the total response data length is 32656 bytes. If this number of bytes is sent on a CEMT response, the CEMT response may be truncated!
4. Errors found during EVESNRSP processing are logged. The following error messages may be displayed:  

```
EVE121E ERROR ON DSIxxx REQUEST IN EVESNRSP, RC = ccc
EVE125E NO STORAGE AVAILABLE ON DSIxxx REQUEST IN EVESNRSP
EVE127E ERROR ON EVESX002 CALL IN EVESNRSP, RC = ccc
EVE141E INCORRECT MQS BUFFER RECEIVED IN EVESNRSP
```



## EVESCCCI - CICS to NetView Communication Interface

Use this subroutine to request, from a CICS transaction, the initiation of a NetView command, a command list, or command processor in NetView. There are three types of EVESCCCI requests:

1. A CONVERSE request, which expects a response from NetView.
2. A SEND request, which does not expect a response from NetView.
3. RESPONSE.

The request is initiated by setting up a parameter list and linking to the EVESCCCI routine, as shown in the following assembler example:

**EXEC CICS LINK PROGRAM(EVESCCCI) COMMAREA(*area*) LENGTH('60')**

The parameter list is located in *area*. Following is an example of a parameter list built for a CONVERSE request (see Table 21 on page 215 for a SEND request parameter list example):

| Table 19. EVESCCCI CONVERSE Request Parameter List |       |        |                                      |
|----------------------------------------------------|-------|--------|--------------------------------------|
| Name                                               | Disp. | Length | Contents and description             |
| OUTIDENT                                           | 000   | 08     | "POUT0001"                           |
| OUTREQID                                           | 008   | 16     | Not used (set by EVESCCCI)           |
| OUTFNAME                                           | 024   | 08     | <i>function</i>                      |
| OUTRTYPE                                           | 032   | 01     | "C"                                  |
|                                                    | 033   | 01     | Reserved (binary zero)               |
| OUTWAITC                                           | 034   | 02     | CONVERSE wait time in seconds        |
| OUTDATAL                                           | 036   | 04     | Length of request data area (binary) |
| OUTDATAA                                           | 040   | 04     | Address of request data area         |

**Note:** Sample copy books for this parameter list are included in the CICS Automation sample library. Refer to "EVEMPINT - EVESCCCI Parameter List Copy Book" on page 216 for field descriptions and important usage information.

If the request data address is located below the line, the response area (if any) is allocated below the line as well. If no data is passed on the CONVERSE request, the length field (OUTDATAL) must be zero. The address field is still used to determine the location of the response area in this case.

If the request is accepted, EVESCCCI sends the request to the NetView EVENTASK optional task, which translates the function into a command list or command processor.

EVESCCCI waits *nn* seconds (see OUTWAITC) for the response to arrive. If the OUTWAITC value is zero, the default wait time is 30 seconds. Valid specifications in the OUTWAITC field range from 1 up to and including 999 (seconds).

If the response does not arrive within the expected interval, the timeout return code is passed to the caller.

EVESCCCI returns the following fields to the caller of the CONVERSE request.

Table 20. EVESCCCI Fields Returned to Caller from CONVERSE Request

| Name     | Disp. | Length | Contents and description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------|-------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OUTRESPA | 044   | 4      | Address of response area                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| OUTRESPL | 048   | 4      | Length of response area                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| OUTRTRNC | 052   | 4      | Binary return code:<br><b>0</b> Successful request<br><b>4</b> Timeout on CONVERSE<br><b>8</b> Program-to-program interface not available (see OUTABNDC for error code)<br><b>12</b> Incorrect parameter list<br><b>16</b> Internal processing error (see OUTABNDC for error code)                                                                                                                                                                                                                                                                     |
| OUTABNDC | 056   | 4      | Error code (character)<br><ul style="list-style-type: none"> <li>OUTRTRNC = 8<br/> <b>C003</b> Program-to-program interface not active<br/> <b>C015</b> CICS LOAD/LINK failure<br/> <b>C017</b> No CICS storage available<br/> <b>C2XX</b> Program-to-program interface request error</li> <li>OUTRTRNC = 16<br/> <b>A...</b> CICS abend codes<br/> <b>C012</b> CICS READQ failure<br/> <b>C016</b> CICS POST failure<br/> <b>C018</b> CICS FREEMAIN failure<br/> <b>C960</b> Incorrect TS item length<br/> <b>C961</b> RQE chain corrupted</li> </ul> |

**Note:** Refer to “EVEMPINT - EVESCCCI Parameter List Copy Book” on page 216 for field descriptions and important usage information.

The response area contains the response (if any) on the CONVERSE request. It may contain a RESPONSE, an ACK, or a NACK. The area has the format as described by the EVEMPINT copy book. **It is the responsibility of the caller of EVESCCCI to free the response area via EXEC CICS FREEMAIN.**

The response area format is similar to the transaction input data passed on a CONVERSE or SEND request from NetView:

| Name     | Disp. | Length | Contents and description                              |
|----------|-------|--------|-------------------------------------------------------|
| INTIDENT | 000   | 08     | “PINT0001”                                            |
| INTREQID | 008   | 16     | <i>domainid  opid</i>                                 |
| INTFNAME | 024   | 08     | <i>function</i>                                       |
| INTRTYPE | 032   | 01     | Response type: “R,” “A,” or “N”                       |
|          | 033   | 03     | Reserved (binary zeros)                               |
| INTDATAL | 036   | 04     | Length of <i>data</i> (binary). Zero for ACK response |

| Name     | Disp. | Length    | Contents and description |
|----------|-------|-----------|--------------------------|
| INTSDATA | 040   | <i>nn</i> | <i>data</i>              |

Errors found by EVESCCCI are returned to the caller via a return code and an error code. When other errors are detected during the processing of a CONVERSE request, CICS Automation returns a NACK response to the CICS transaction. The NACK response data indicates the type of error that occurred. The NACK response is logged via message EVE180E. The following NACK responses can be expected:

EVE180E text

```

EVE121E ERROR ON DISxxx REQUEST IN progame, RC = ccc
EVE122E tttttttt TASK NOT ACTIVE
EVE125E NO STORAGE AVAILABLE ON DSIXxx REQUEST IN progame
EVE142E FUNCTION funcname NOT FOUND IN memname

```

Following is an example of a parameter list built by a SEND request:

| Table 21. EVESCCCI SEND Request Parameter List |       |        |                                      |
|------------------------------------------------|-------|--------|--------------------------------------|
| Name                                           | Disp. | Length | Contents and description             |
| OUTIDENT                                       | 000   | 08     | "POUT0001"                           |
| OUTREQID                                       | 008   | 16     | Not used (set by EVESCCCI)           |
| OUTFNAME                                       | 024   | 08     | <i>function</i>                      |
| OUTRTYPE                                       | 032   | 01     | "S"                                  |
|                                                | 033   | 01     | Reserved (binary zero)               |
|                                                | 034   | 02     | Not used for SEND (binary zeros)     |
| OUTDATAL                                       | 036   | 04     | Length of request data area (binary) |
| OUTDATAA                                       | 040   | 04     | Address of request data area         |

**Note:** Refer to "EVEMPINT - EVESCCCI Parameter List Copy Book" on page 216 for field descriptions and important usage information.

If no data is passed on the SEND request, the length field (OUTDATAL) must be 0 (zero).

On a SEND request, errors found by EVESCCCI are returned to the caller via a return code and an error code. Other errors detected during the processing of a SEND request are not returned to the requestor. The errors are only logged.

## EVEMPINT - EVESCCCI Parameter List Copy Book

The EVEMPINT copy book describes the CICS Automation program-to-program interface parameter list at the CICS side. It consists of two parts: one part describes the input data for CICS transactions and the other part describes the format of output requests from CICS transactions.

The following data area is passed to a started CICS transaction as result of a NetView CONVERSE or SEND request. The same data area is passed in the response area as result of a NetView RESPONSE, ACK, or NACK response. See "EVESNCCI - NetView to CICS Communication Interface" on page 207.

**Note:** The started transaction must obtain the input data using an EXEC CICS RETRIEVE command.

| Name     | Disp. | Length    | Contents and description                                                                                                                                                                                                                                                                                                                                                                                 |
|----------|-------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| INTIDENT | 000   | 08        | Provides an eye catcher and a block format level.                                                                                                                                                                                                                                                                                                                                                        |
| INTREQID | 008   | 16        | This field contains the request identifier which is used to relate a response to a specific request. For a CONVERSE request from NetView, it contains the <i>domainid//opid</i> concatenation specified on the EVESNCCI command. For responses from NetView, it contains the request identifier allocated by the EVESCCCI routine on a CICS CONVERSE request. This field is not used for a SEND request. |
| INTFNAME | 024   | 08        | Contains the function name specified with the EVESNCCI FUNCTION= keyword.                                                                                                                                                                                                                                                                                                                                |
| INTRTYPE | 032   | 01        | The response type: C for CONVERSE, S for SEND, R for RESPONSE, A for ACK, and N for NACK.                                                                                                                                                                                                                                                                                                                |
|          | 033   | 03        | Reserved. Binary zeros.                                                                                                                                                                                                                                                                                                                                                                                  |
| INTDATAL | 036   | 04        | Contains the length of the request or response data. This field may have a 0 (zero) value. For an ACK response, this field is 0 (zero).                                                                                                                                                                                                                                                                  |
| INTSDATA | 040   | <i>nn</i> | The request or response data, if any. The length of the response data is contained in the INTDATAL field.                                                                                                                                                                                                                                                                                                |

The following fields are set by the caller of the EVESCCCI routine.

| Name     | Disp. | Length | Contents and description                                                                                                                                                                                                                                                                                                                                             |
|----------|-------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OUTIDENT | 000   | 08     | Provides an eye catcher and a block format level. Must be set to POUT0001.                                                                                                                                                                                                                                                                                           |
| OUTREQID | 008   | 16     | This field is used as a request identifier to relate a response to a specific CONVERSE request. For CICS CONVERSE requests this field is set by the EVESCCCI routine. For CICS responses, the field must be set by the caller (copied from INTREQID). This field is not used for SEND requests.                                                                      |
| OUTFNAME | 024   | 08     | Specifies the function name. The EVENTASK initialization member must contain a SERVER=REQUEST entry (for CONVERSE or SEND) or a SERVER=RESPONSE entry (for RESPONSE) specification. For responses, this field is normally copied from the transaction input data (INTFNAME field)<br><br><b>Note:</b> If the name is less than eight characters, pad it with blanks. |

## EVEMPINT - EVESCCCI Parameter List Copy Book

| Name     | Disp. | Length | Contents and description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------|-------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OUTRTYPE | 032   | 01     | Specifies the type of command as REQUEST, SEND, RESPONSE, ACK, or NACK. REQUEST (also referred to as CONVERSE) and SEND are requests. RESPONSE, ACK, and NACK are responses.<br><br>REQUEST starts a command processor or a command list in NetView. A response is expected. SEND also starts a command processor or a command list in NetView, but no response is expected. RESPONSE is used to send data to a NetView task. ACK signals the successful completion of a CONVERSE request. No data is provided. NACK signals the unsuccessful completion of a CONVERSE request. Data may optionally be provided (maximum of 100 bytes). It is recommended for health checking. |
|          | 033   | 01     | Reserved. Binary zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| OUTWAITC | 034   | 02     | Specifies the number of seconds (binary value) to wait for a response on a CONVERSE request. If the response does not arrive within the specified time interval, the timeout return code is passed to the caller. If binary zero is specified, the default wait interval (30 seconds) is used. The maximum binary value that can be specified is 999. The field must contain binary zeros for all requests other than CONVERSE.                                                                                                                                                                                                                                                |
| OUTDATAL | 036   | 04     | This field must be set to the length of the CONVERSE or SEND request data area or to the length of the RESPONSE or NACK response area. The maximum length of a CONVERSE or SEND request area or a RESPONSE response area is 32656 bytes. The maximum length of a NACK response area is 100 bytes.                                                                                                                                                                                                                                                                                                                                                                              |
| OUTDATAA | 040   | 04     | Specifies the address of the CONVERSE or SEND request area or the address of the RESPONSE or NACK response area. If the request data address is located below the line, the response area (if any) is allocated below the line as well. If no data is passed on the CONVERSE request, the length field (OUTDATAL) must be zero. The address field is still used to determine the location of the response area in this case.                                                                                                                                                                                                                                                   |

The following fields are set by EVESCCCI upon return to the caller:

| Name     | Disp. | Length | Contents and description                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|----------|-------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OUTRESPA | 044   | 04     | Address of response area allocated by EVESCCCI. It may contain a RESPONSE, ACK, or NACK response. This field is only returned on a successful CONVERSE request (OUTRTRNC=0). <b>It is the responsibility of the caller of EVESCCCI to free the response area using EXEC CICS FREEMAIN.</b>                                                                                                                                                                                                  |
| OUTRESPL | 048   | 04     | Contains the length of the response area addressed by the OUTRESPA field.                                                                                                                                                                                                                                                                                                                                                                                                                   |
| OUTRTRNC | 052   | 04     | Contains the return code which will have one of the following binary values:<br><br><b>000</b> Successful request.<br><b>004</b> Timeout on CONVERSE.<br><b>008</b> Program-to-program interface not available (see OUTABNDC for error codes). A transaction dump is provided depending on the error code.<br><b>012</b> Incorrect parameter list. No transaction dump is provided.<br><b>016</b> Internal processing error (see OUTABNDC for error codes). A transaction dump is provided. |

## EVEMPINT - EVESCCCI Parameter List Copy Book

| Name     | Disp. | Length | Contents and description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------|-------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OUTABNDC | 056   | 04     | <p>This field contains an error code for return codes 008 and 016. For return code 008 the field will have one of the following character values:</p> <p><b>C003</b> The CICS component of the program-to-program interface is not active. Use the COPS transaction to start it. No transaction dump is provided. An error message is logged.</p> <p><b>C015</b> A CICS LOAD of or LINK to a required module was not successful. Ensure that EVESPERR and EVESPMMSG have been properly installed and are enabled. A transaction dump is provided. An error message is logged.</p> <p><b>C017</b> No CICS storage is available. No transaction dump is provided.</p> <p><b>C2xx</b> A program-to-program interface request error occurred, where xx contains the program-to-program interface request return code. For error codes C220, C222, C223, C225, C231, C233, C236, C240, and C290, a transaction dump is provided and an error message is logged.</p> <p>For return code 016 the field will have one of the following character values:</p> <p><b>A***</b> A CICS abend has occurred. A transaction dump is provided. An error message is logged for most A*** errors.</p> <p><b>C012</b> An unexpected error has occurred on a READQ command. A transaction dump is provided. An error message is logged.</p> <p><b>C016</b> An unexpected error has occurred on a POST command. A transaction dump is provided. An error message is logged.</p> <p><b>C018</b> An unexpected error has occurred on a FREEMAIN command. A transaction dump is provided.</p> <p><b>C961</b> Internal error. Incorrect TS item length. A transaction dump is provided. An error message is logged.</p> <p><b>C962</b> Internal error. The RQE chain is corrupted. A transaction dump is provided. An error message is logged.</p> |

When a RESPONSE response (R) is sent, *function* is used to locate the name of a command processor or command list in the EVENTASK initialization member. This command is scheduled under a task (also defined in the initialization member) with the following command text:

| Name     | Disp. | Length | Contents and description                                            |
|----------|-------|--------|---------------------------------------------------------------------|
|          | 000   | 8      | Command processor or command list name                              |
|          | 008   | 8      | " " (8 blanks)                                                      |
| RHDCRVID | 016   | 8      | Program-to-program interface receiver identification                |
| RHDSNDID | 024   | 8      | Generic applid (Program-to-program interface sender identification) |
| RHDPRCNM | 032   | 8      | Program-to-program interface sender's JOB- or STC-name              |
| RHDDOMID | 040   | 8      | <i>domainid</i>                                                     |
| RHDTSKID | 048   | 8      | <i>opid</i>                                                         |
| RHDFNAME | 056   | 8      | <i>function</i>                                                     |
| RHDRTYPE | 064   | 1      | "R"                                                                 |
|          | 065   | 7      | "*****" (7 asterisks)                                               |
| RHDSDATA | 072   | n      | Response data (n = OUTDATAL)                                        |

A common response processor, EVESNRSP, is available that turns the response data into a multi-line WTO (EVE79xl) which is sent to *opid* in *domainid*.

## Examples of Usage

### Example 1

This assembler example shows the processing of a CICS CONVERSE request.

```

* ISSUE A CONVERSE REQUEST *

*
* XC CISPOUTP,CISPOUTP ZERO PARAMETER LIST
* LA R6,CISPOUTP ADDRESS PARAMETER LIST
* USING OUTDSECT,R6 ESTABLISH ADDRESSABILITY
* SPACE 1
* MVC OUTIDENT,=C'POUT0001' SET EYE CATCHER/BLOCK LEVEL
* MVC OUTFNAME,=CL8'TESTCLST' SET FUNCTION NAME
* MVI OUTRTYPE,OUTRTYPE SET CONVERSE REQUEST TYPE
* LA R1,L'CONVTEXT LENGTH OF VARIABLE DATA
* ST R1,OUTDATAL SET LENGTH OF VARIABLE DATA
* LA R1,CONVTEXT ADDRESS OF VARIABLE DATA
* ST R1,OUTDATAA SET ADDRESS OF CONVERSE DATA
* MVC CISHWORD,=Y(OUTHL) LENGTH OF PARAMETER LIST
*
* EXEC CICS LINK, REQUEST PROGRAM LINK, TO *
* PROGRAM('EVESCCCI'), CPDS COMMUNICATION INTERFACE *
* COMMAREA(CISPOUTP), PARAMETER LIST *
* LENGTH(CISHWORD) PARAMETER LIST LENGTH *
*
* L R15,OUTTRNC PICK UP THE RETURN CODE
* Process the return code and error code please!
*
* L R4,OUTRESPA ADDRESS RESPONSE AREA
* USING INTDSECT,R4 ESTABLISH ADDRESSABILITY
*
* LA R14,INTSDATA ADDRESS RESPONSE DATA
* L R15,INTDATAL LENGTH RESPONSE DATA
* Do some meaningful processing please!
*
* CONVTEXT DC C'CONVERSE TEXT FROM CICS'
*
* DFHEISTG ,
*
* CISPOUTP DS CL(OUTHL) RESPONSE PARAMETER LIST
* CISHWORD DS H PARAMETER BLOCK LENGTH
*
* DFHEIEND ,
*
* EVEMPINT , DEFINE INTERFACE DSECT
```

## EVEMPINT - EVESCCCI Parameter List Copy Book

### Example 2

This assembler example shows the processing of a NetView CONVERSE request in CICS.

```

* RETRIEVE TRANSACTION INPUT DATA *

*
* EXEC CICS RETRIEVE, GET INPUT DATA *
* SET(R4), RETURN ADDRESS HERE *
* LENGTH(CISHWORD) RETURN LENGTH HERE
*
* USING INTDSECT,R4 ESTABLISH ADDRESSABILITY
*
* LA R14,INTSDATA ADDRESS REQUEST DATA
* L R15,INTDATAL LENGTH REQUEST DATA
* Do some meaningful processing please!
*

* RETURN A 'NORMAL' RESPONSE *

*
* XC CISPOUTP,CISPOUTP ZERO PARAMETER LIST
* LA R6,CISPOUTP ADDRESS PARAMETER LIST
* USING OUTDSECT,R6 ESTABLISH ADDRESSABILITY
*
* MVC OUTIDENT,=C'POUT0001' SET EYE CATCHER/BLOCK LEVEL
* MVC OUTREQID,INTREQID SET REQUEST IDENTIFIER
* MVC OUTFNAME,INTFNAME SET FUNCTION NAME
* MVI OUTRTYPE,OUTRTYPR SET RESPONSE RESPONSE TYPE
* LA R1,L'RESPTEXT LENGTH OF VARIABLE DATA
* ST R1,OUTDATAL SET LENGTH OF VARIABLE DATA
* LA R1,RESPTEXT ADDRESS OF VARIABLE DATA
* ST R1,OUTDATAA SET ADDRESS OF RESPONSE DATA
* MVC CISHWORD,=Y(OUTHL) LENGTH OF PARAMETER LIST
*
* EXEC CICS LINK, REQUEST PROGRAM LINK, TO
* PROGRAM('EVESCCCI'), CPDS COMMUNICATION INTERFACE
* COMMAREA(CISPOUTP), PARAMETER LIST
* LENGTH(CISHWORD) PARAMETER LIST LENGTH
*
* L R15,OUTRTRNC PICK UP THE RETURN CODE
* Process the return code and error code please!
*
* RESPTEXT DC C'RESPONSE TEXT FROM CICS'
*
* DFHEISTG ,
*
* CISPOUTP DS CL(OUTHL) RESPONSE PARAMETER LIST
* CISHWORD DS H LENGTH OF PARAMETER BLOCK
*
* DFHEIEND ,
*
* EVEMPINT , DEFINE INTERFACE DSECT

```



## Appendix B. Sample JCL

This chapter shows examples of the JCL.

### EVESJ010—Copy the Sample Members to DSIPARM

```
//EVESJ010 JOB 'ACCOUNTING INFORMATION','COPY DSIPARM MEMBERS', 00010000
// CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1) ,TYPRUN=HOLD 00020000
//***** 00030000
//* (C) COPYRIGHT IBM CORP. 1990,1995 - ALL RIGHTS RESERVED * 00040000
//* * 00050000
//* LICENSED MATERIALS - PROPERTY OF IBM * 00060000
//* RESTRICTED MATERIALS OF IBM * 00070000
//* 5685-151 * 00080000
//* * 00090000
//* US GOVERNMENT USERS RESTRICTED RIGHTS - * 00100000
//* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY * 00110000
//* GSA ADP SCHEDULE CONTRACT WITH IBM CORP. * 00120000
//***** 00130000
//* APAR# DATE * 00140000
//* ----- * 00150000
//* * 00160000
//***** 00170000
//* * 00180000
//* DESCRIPTION: JCL TO COPY AOC/CICS SAMPLIB MEMBERS TO THE NETVIEW * 00190000
//* DSIPARM DATASET * 00200000
//* * 00210000
//***** 00220000
//* NOTE 1) CHANGE Q1 TO MATCH THAT OF THE ACTUAL HIGH LEVEL QUALIFIER 00230000
//* USED TO ALLOCATE THE AOC/CICS DATA SETS 00240000
//* 2) UPDATE THE DSIPARM DD STATEMENT TO POINT TO THE DSIPARM 00250000
//* DATASET WHERE AOC/CICS WILL BE TESTED 00260000
//* 3) NORMAL JOB CONDITION CODE IS ZERO(0). 00270000
//* * 00280000
//***** 00290000
//COPYPARM PROC Q1='AOCCICS.V1R4M0' 00300000
//LOADCOPY EXEC PGM=IEBCOPY,REGION=512K 00310000
//SYSPRINT DD SYSOUT=* 00320000
//***** 00330000
//* COPY SAMPLIB MEMBERS TO THE DSIPARM DATASET * 00340000
//***** 00350000
//SEVESAMP DD DSN=&Q1..SEVESAMP,DISP=SHR 00360000
//DSIPARM DD DSN=CNM.SA01.DSIPARM,DISP=SHR 00370000
//SYSUT2 DD UNIT=SYSDA,SPACE=(CYL,(1,1)) 00380000
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,1)) 00390000
// PEND 00400000
//COPY EXEC COPYPARM 00410000
//SYSIN DD * 00420000
COPY INDD=((SEVESAMP,R)),OUTDD=DSIPARM 00430001
SELECT MEMBER=(EVECFG01,EVECFG) 00440003
SELECT MEMBER=(EVECMD,EVESECUR) 00441003
SELECT MEMBER=EVEOPF 00450003
SELECT MEMBER=EVEDMN 00450103
SELECT MEMBER=EVENTASK 00451003
SELECT MEMBER=EVED0001 00460000
SELECT MEMBER=(EVEDACBA,EVEDACBB,EVEDACB1,EVEDACB2) 00470000
SELECT MEMBER=(EVEDAUIA,EVEDAUIB,EVEDAUI1,EVEDAUI2) 00480000
SELECT MEMBER=(EVEDHLTA,EVEDHLTB,EVEDHLT1,EVEDHLT2) 00490000
SELECT MEMBER=(EVEDLMTA,EVEDLMTB,EVEDLMT1,EVEDLMT2) 00500000
SELECT MEMBER=(EVEDMSGA,EVEDMSGB,EVEDMSG1,EVEDMSG2) 00510000
SELECT MEMBER=(EVEDSTGA,EVEDSTGB,EVEDSTG1,EVEDSTG2) 00520000
```

|        |                                               |          |
|--------|-----------------------------------------------|----------|
| SELECT | MEMBER=(EVEDTIMA,EVEDTIMB,EVEDTIM1,EVEDTIM2)  | 00530000 |
| SELECT | MEMBER=(EVEDTRNA,EVEDTRNB,EVEDTRN1,EVEDTRN2)  | 00540000 |
| SELECT | MEMBER=(EVEESA01,EVEESA02,EVEESA03,EVEESA04)  | 00550000 |
| SELECT | MEMBER=((EVEMSG00,CICSMG0),EVEMCON0,EVEMEVE0) | 00570000 |
| SELECT | MEMBER=((EVEMSG01,CICSMG1))                   | 00580000 |
| SELECT | MEMBER=(EVEMDFH1,EVEMCON1,EVEMEVE1)           | 00590000 |
| /**    |                                               | 00600000 |

## EVESJ015—CICS/MVS Definitions

```

//EVESJ015 JOB 'ACCOUNTING INFORMATION','CSD DEFINE', 00010000
// CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1) ,TYPRUN=HOLD 00020000
//***** 00030000
//* (C) COPYRIGHT IBM CORP. 1990,1995 - ALL RIGHTS RESERVED * 00040001
//* 00050000
//* LICENSED MATERIALS - PROPERTY OF IBM * 00060000
//* RESTRICTED MATERIALS OF IBM * 00070000
//* 5685-151 * 00080000
//* 00090000
//* US GOVERNMENT USERS RESTRICTED RIGHTS - * 00100000
//* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY * 00110000
//* GSA ADP SCHEDULE CONTRACT WITH IBM CORP. * 00120000
//***** 00130000
//* APAR# DATE * 00140000
//* ----- * 00150000
//* $01=0W32834,V1R4,10AUG98,APC(GJD): SUPPORT CICS V5.1 AND V5.2 * 00160000
//* 00160000
//***** 00170000
//* DESCRIPTION: Add AOC/CICS definitions to CICS CSD * 00180000
//* 00190000
//* NOTE: 1) UPDATE STEPLIB DD AS FOLLOWS: * 00200000
//* a) CICSxxx.SDFHLOAD should point to the CICS library * 00210000
//* defined during the installation of CICS. * 00220000
//* - In CICS 2.1.2, the low-level qualifier is LOADLIB * 00230000
//* b) aoccics.v1r4m0.SEVEMOD2 should be changed to match the * 00240001
//* qualifier selected during AOC/CICS installation * 00250000
//* 2) Change name of DFHCSD data set to match the actual CSD * 00260000
//* data set to be modified. * 00270000
//* 3) Change the UPDGRADE(EVECDxx) statement to reflect the * 00280000
//* appropriate CICS level you are installing AOC/CICS on. * 00290000
//* - EVECD21 for CICS 2.1.2 * 00300000
//* - EVECD32 for CICS 3.2.1 * 00310000
//* - EVECD33 for CICS 3.3 * 00320000
//* - EVECD41 for CICS 4.1 * 00330000
//* - EVECD51 for CICS 5.1 * 00330000
//* - EVECD52 for CICS 5.2 * 00330000
//* 4) Change the ADD GROUP statements, LIST(yourlist) to add * 00340000
//* the new AOC/CICS groups to your CSD group lists. * 00350000
//* 00360000
//* CAUTION: See instructions in the AOC/CICS Programmer's & INSTALL * 00370000
//* guide before submitting this job. This job should be * 00380000
//* run for each unique CSD dataset and CICS level that you * 00390000
//* have in your environment. * 00400000
//*****/ 00410000
//* */ 00420000
//UPGRADE EXEC PGM=DFHCSDUP 00430000
//STEPLIB DD DISP=SHR,DSN=CICSxxx.SDFHLOAD 00440000
// DD DISP=SHR,DSN=aoccics.v1r4m0.SEVEMOD2 00450001
//DFHCSD DD DISP=SHR,DSN=yourcsd.DFHCSD 00460000
//SYSUT1 DD UNIT=3380,SPACE=(1024,(100,100)) 00470000
//SYSPRINT DD SYSOUT=* 00480000
//SYSIN DD * 00490000
UPGRADE USING(EVECDxx) 00500000
ADD GROUP(EVEGRP1) LIST(yourlist) 00510000
ADD GROUP(EVEGRP2) LIST(yourlist) 00520000
ADD GROUP(EVEGRP3) LIST(yourlist) 00530000
/*

```

## EVESJ020—Assemble and Link Edit Definition Members

```

EVESJ020 JOB 'ACCOUNTING INFORMATION','USER TABLES', 00010000
// CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1) 00020000
//***** 00030000
//* (C) COPYRIGHT IBM CORP. 1990,1994 - ALL RIGHTS RESERVED *
//* *
//* LICENSED MATERIALS - PROPERTY OF IBM *
//* RESTRICTED MATERIALS OF IBM *
//* 5685-151 *
//* *
//* US GOVERNMENT USERS RESTRICTED RIGHTS - *
//* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY *
//* GSA ADP SCHEDULE CONTRACT WITH IBM CORP. *
//*****
//* APAR# DATE DESCRIPTION * 00080000
//* -----
//* $02=0W32834,V1R4,14MAY98,APC(AVF): Support for CICS V5R1 & V5R2. *
//* $01=PN52486 03/11/94 MP CICS AO SUPPORT FOR CICS V4 *
//* PN175765 MP 07/06/92 CICS V3R3 SUPPORT * 00090000
//*****
//* DESCRIPTION: THIS JOB ASSEMBLES AND LINK-EDITS USER MAINTAINED * 00060000
//* EVESCMT2, EVESCMT3, EVESPINM TABLES. * 00070000
//***** 00100000
//* NOTE 1) UPDATE DATASET NAMES, VOL=SER, AND UNIT FIELDS. * 00110000
//* 2) IF NECESSARY, UPDATE THE SOURCE AND LOADLIB DATASET NAMES * 00120000
//* 3) VERIFY THAT BLKSIZE OF FIRST DATA SET OF SYSLIB CONCAT * 00160000
//* IN ASM STEP IS LARGEST. * 00161000
//* 4) UPDATE LIB2 PARAMETER TO POINT TO ACTUAL USER LINK LIB * 00162000
//***** 00200000
//CICSLNK PROC Q1='AOCCICS.V1R3M0', * AOC/CICS HIGH LEVEL QUALIFIER 00232000
// LIB1='AOCCICS.V1R3M0.SEVESRC', * INPUT SOURCE LIBRARY 00232000
// NAME1=, * INPUT SOURCE NAME 00232000
// LIB2='CICS.USERLNK', * LINK-EDIT OUTPUT LIBRARY 00232000
// NAME2=, * LINK-EDIT OUTPUT NAME 00232000
// LNKIN= * LINK-EDIT CONTROL CARDS 00232000
//***** 00255300
//* ASSEMBLY STEP * 00256000
//***** 00260000
//ASM EXEC PGM=IEV90, 00270000
// REGION=3M, 00280000
// PARM='DECK,NOBJECT,LIST,ALIGN,BATCH' 00290000
//SYSLIB DD DSN=&Q1..SEVESRC,DISP=SHR 00320000
// DD DSN=SYS1.MACLIB,DISP=SHR 00330000
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1)) 00350000
//SYSUT2 DD UNIT=SYSDA,SPACE=(CYL,(1,1)) 00360000
//SYSUT3 DD UNIT=SYSDA,SPACE=(CYL,(1,1)) 00370000
//SYSPUNCH DD DSN=&&LOADSET, 00380000
// UNIT=SYSDA,DISP=(,PASS), 00390000
// SPACE=(400,(100,100,1)) 00400000
//SYSPRINT DD SYSOUT=X 00410000
//SYSIN DD DSN=&LIB1(&NAME1),DISP=SHR 00420000
//***** 00430000
//* GET LINK-EDIT CONTROL STATEMENTS * 00440000
//***** 00440100
//COPYLINK EXEC PGM=IEBGNER 00441000
//SYSUT1 DD DSN=&Q1..SEVESAMP(&LNKIN),DISP=SHR 00442000
//SYSUT2 DD DSN=&©LINK,DISP=(NEW,PASS), 00443000
// DCB=(LRECL=80,BLKSIZE=400,RECFM=FB), 00444000
// UNIT=SYSDA,SPACE=(400,(20,20)) 00445000
//SYSPRINT DD SYSOUT=X 00446000
//SYSIN DD DUMMY 00447000
//***** 00450000
//* LKED * 00540000

```

|                                                         |          |
|---------------------------------------------------------|----------|
| //*****                                                 | 00550000 |
| //LKED EXEC PGM=IEWL,PARM='RENT,MAP,XREF,LIST',         | 00560000 |
| // COND=(8,LE,ASM),REGION=512K                          | 00570000 |
| //SYSLIB DD DSN=&Q1..SEVEMOD2,DISP=SHR                  | 00590000 |
| //SYSLMOD DD DSN=&LIB2(&NAME2),DISP=SHR                 | 00620000 |
| //SYSUT1 DD DSN=&&SYSUT1,UNIT=SYSDA,                    | 00630000 |
| // SPACE=(1024,(200,50),,CONTIG,ROUND),DCB=BLKSIZE=1024 | 00640000 |
| //SYSPRINT DD SYSOUT=X                                  | 00650000 |
| //SYSLIN DD DSN=&&LOADSET,DISP=(OLD,DELETE)             | 00670000 |
| // DD DSN=&&COPYLINK,DISP=(OLD,DELETE)                  | 00671000 |
| // PEND                                                 | 00680000 |
| //EVESCM2 EXEC CICSLNK,NAME1=EVESCM2,NAME2=EVESCM2,     | 00690000 |
| // LNKIN=EVELCME2                                       | 00691000 |
| //EVESCM3 EXEC CICSLNK,NAME1=EVESCM3,NAME2=EVESCM3,     | 00700000 |
| // LNKIN=EVELCME3                                       | 00710000 |
| //EVESCM33 EXEC CICSLNK,NAME1=EVESCM3,NAME2=EVESCM33,   | 00700000 |
| // LNKIN=EVELCM33                                       | 00710000 |
| //EVESCM41 EXEC CICSLNK,NAME1=EVESCM3,NAME2=EVESCM41,   | 00700000 |
| // LNKIN=EVELCM41                                       | 00710000 |
| //EVESCM51 EXEC CICSLNK,NAME1=EVESCM3,NAME2=EVESCM51,   | 00700000 |
| // LNKIN=EVELCM51                                       | 00710000 |
| //EVESCM52 EXEC CICSLNK,NAME1=EVESCM3,NAME2=EVESCM52,   | 00700000 |
| // LNKIN=EVELCM52                                       | 00710000 |
| //EVESPINM EXEC CICSLNK,NAME1=EVESPINM,NAME2=EVESPINM,  | 00720000 |
| // LNKIN=EVELPINM                                       | 00730000 |

## EVESJ022—Assemble and Link Edit Health Check Programs

```

//EVESJ022 JOB 'ACCOUNTING INFORMATION','HEALTH/TRANS', 00010000
// CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1) 00020000
//***** 00030000
//* (C) COPYRIGHT IBM CORP. 1990,1995 - ALL RIGHTS RESERVED * 00030100
//* * 00030200
//* LICENSED MATERIALS - PROPERTY OF IBM * 00030300
//* RESTRICTED MATERIALS OF IBM * 00030400
//* 5685-151 * 00030500
//* * 00030600
//* US GOVERNMENT USERS RESTRICTED RIGHTS - * 00030700
//* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY * 00030800
//* GSA ADP SCHEDULE CONTRACT WITH IBM CORP. * 00031000
//***** 00040000
//* APAR# DATE * 00042000
//* ----- ----- * 00043000
//* * 00043100
//***** 00043200
//* * 00044000
//* DESCRIPTION: SAMPLE JCL FOR HEALTH CHECK PROGRAM AND TRANSACTION * 00050000
//* DEFINITIONS * 00051000
//* * 00052000
//* NOTE: * 00053000
//* 1) UPDATE Q1 TO MATCH ALIAS USED FOR AOC/CICS DATASETS * 00054000
//* 2) UPDATE Q2 & Q3 AS APPROPRIATE FOR YOUR SYSTEM * 00054100
//* 2) MAKE SURE STEPLIB POINTS TO CORRECT DATA SET * 00055000
//* 4) UPDATE LIB1 STATEMENT TO POINT TO YOUR SOURCE INPUT * 00057000
//* LIBRARY * 00057100
//* 5) UPDATE LIB2 STATEMENT TO POINT TO USER LINK FOR CICS * 00057200
//* * 00057300
//* WARNING: JCL USES CICS V3 LIBRARY NAMES. IF COMPILING ON * 00057400
//* CICS 2.1.2, COMPARE THIS PROC TO THE THE SUPPLIED * 00057500
//* DFHEITVL AND CORRECT LIBRARY NAMES AS APPROPRIATE. * 00057600
//***** 00140000
//DFHEITVL PROC SUFFIX=1$, * CICS TRANSLATOR SUFFIX 00150000
// Q1='AOCCICS.V1R4M0', * AOC/CICS HIGH LEVEL QUALIFIER 00160000
// Q2='CICS330', * HLQ FOR THE CICS LIBRARIES 00160100
// Q3='SYS1', * HLQ FOR THE COBOL2 LIBRARIES 00160200
// LIB1='AOCCICS.V1R4M0.SEVESRC', * INPUT SOURCE LIBRARY 00160300
// NAME1=, * INPUT SOURCE NAME 00160400
// LIB2='CICS.USERLNK', * LINK-EDIT OUTPUT LIBRARY 00160500
// NAME2=, * LINK-EDIT OUTPUT NAME 00160600
// STUBS=DFHEILIC, * CICS COBOL STUB MODULE 00161000
// OUTC=A, * JES OUTPUT CLASS 00180000
// REG=2048K, * SIZE FOR TRANSLATE,COMPILE,LINK 00190000
// LNKPARM='(LIST,XREF)', * LINK-EDIT PARAMETERS 00200000
// WORK=SYSDA * UNIT STATEMENT FOR DD ALLOCATIONS 00210000
//* 00220000
//* THIS PROCEDURE CONTAINS 3 STEPS 00230000
//* 1. EXEC THE COBOL TRANSLATOR 00240000
//* (USING THE SUPPLIED SUFFIX 1$) 00250000
//* 2. EXEC THE VS COBOL II COMPILER 00260000
//* 3. LINKEDIT THE OUTPUT TO CICS211.LOADLIB 00270000
//* 00280000
//* NOTE: FOR OS/VS COBOL USE PROCEDURE DFHEITCL 00290000
//* 00300000
//* THE FOLLOWING JCL SHOULD BE USED 00310000
//* TO EXECUTE THIS PROCEDURE 00320000
//* 00330000
//* //APPLPROG EXEC DFHEITVL 00340000
//* //TRN.SYSIN DD * 00350000
//* . 00360000
//* . APPLICATION PROGRAM 00370000

```

|            |                                                         |          |
|------------|---------------------------------------------------------|----------|
| //*        | .                                                       | 00380000 |
| //*        | /*                                                      | 00390000 |
| //*        | //LKED.SYSIN DD *                                       | 00400000 |
| //*        | NAME ANYNAME(R)                                         | 00410000 |
| //*        | /*                                                      | 00420000 |
| //*        |                                                         | 00430000 |
| //*        | WHERE ANYNAME IS THE NAME OF YOUR APPLICATION PROGRAM   | 00440000 |
| //*        |                                                         | 00450000 |
| //TRN      | EXEC PGM=DFHECP&SUFFIX,                                 | 00460000 |
| //         | PARM='COBOL2',                                          | 00470000 |
| //         | REGION=&REG                                             | 00480000 |
| //STEPLIB  | DD DSN=&Q2..SDFHLOAD,DISP=SHR                           | 00490000 |
| //SYSPRINT | DD SYSOUT=&OUTC                                         | 00500000 |
| //SYSPUNCH | DD DSN=&SYSCIN,                                         | 00510000 |
| //         | DISP=(,PASS),UNIT=&WORK,                                | 00520000 |
| //         | DCB=BLKSIZE=400,                                        | 00530000 |
| //         | SPACE=(400,(400,100))                                   | 00540000 |
| //SYSIN    | DD DSN=&LIB1(&NAME1),DISP=SHR                           | 00541000 |
| //COB      | EXEC PGM=IGYCRCTL,REGION=&REG,                          | 00550000 |
| //         | PARM='NODYNAM,LIB,OBJECT,RENT,RES,APOST,MAP,XREF'       | 00560000 |
| //STEPLIB  | DD DSN=&Q3..COB2COMP,DISP=SHR                           | 00570000 |
| //SYSLIB   | DD DSN=&Q2..SDFHCOB,DISP=SHR                            | 00580000 |
| //         | DD DSN=&Q2..SDFHSAMP,DISP=SHR                           | 00590000 |
| //         | DD DSN=&Q2..SDFHMAC,DISP=SHR                            | 00591000 |
| //         | DD DSN=&Q1..SEVESRC,DISP=SHR                            | 00600000 |
| //SYSPRINT | DD SYSOUT=&OUTC                                         | 00610000 |
| //SYSIN    | DD DSN=&SYSCIN,DISP=(OLD,DELETE)                        | 00620000 |
| //SYSLIN   | DD DSN=&LOADSET,DISP=(MOD,PASS),                        | 00630000 |
| //         | UNIT=&WORK,SPACE=(80,(250,100))                         | 00640000 |
| //SYSUT1   | DD UNIT=&WORK,SPACE=(460,(350,100))                     | 00650000 |
| //SYSUT2   | DD UNIT=&WORK,SPACE=(460,(350,100))                     | 00660000 |
| //SYSUT3   | DD UNIT=&WORK,SPACE=(460,(350,100))                     | 00670000 |
| //SYSUT4   | DD UNIT=&WORK,SPACE=(460,(350,100))                     | 00680000 |
| //SYSUT5   | DD UNIT=&WORK,SPACE=(460,(350,100))                     | 00690000 |
| //COPYLINK | EXEC PGM=IEBGENER,COND=(7,LT,COB)                       | 00700000 |
| //SYSUT1   | DD DSN=&Q2..SDFHCOB(&STUBS),DISP=SHR                    | 00700100 |
| //SYSUT2   | DD DSN=&&COPYLINK,DISP=(NEW,PASS),UNIT=&WORK,           | 00700200 |
| //         | DCB=(LRECL=80,BLKSIZE=400,RECFM=FB),SPACE=(400,(20,20)) | 00700300 |
| //SYSPRINT | DD SYSOUT=&OUTC                                         | 00700400 |
| //SYSIN    | DD DUMMY                                                | 00700500 |
| //LKED     | EXEC PGM=IEWL,REGION=&REG,                              | 00701000 |
| //         | PARM=&LNKPARM,COND=(5,LT,COB)                           | 00710000 |
| //SYSLIB   | DD DSN=&Q2..SDFHLOAD,DISP=SHR                           | 00720000 |
| //         | DD DSN=&Q3..COB2CICS,DISP=SHR                           | 00730000 |
| //         | DD DSN=&Q3..COB2LIB,DISP=SHR                            | 00740000 |
| //SYSLMOD  | DD DSN=&LIB2(&NAME2),DISP=SHR                           | 00750000 |
| //SYSUT1   | DD UNIT=&WORK,DCB=BLKSIZE=1024,                         | 00760000 |
| //         | SPACE=(1024,(200,20))                                   | 00770000 |
| //SYSPRINT | DD SYSOUT=&OUTC                                         | 00780000 |
| //SYSLIN   | DD DSN=&&COPYLINK,DISP=(OLD,DELETE)                     | 00790000 |
| //         | DD DSN=&&LOADSET,DISP=(OLD,DELETE)                      | 00800000 |
| //         | PEND                                                    | 00820000 |
| //         | EXEC DFHEITVL,NAME1=EVECHLTH,NAME2=EVECHLTH             | 00830000 |
| //*        |                                                         | 00840000 |





## Appendix C. Sample and Source

This section contains the samples and source that you can use to customize your automation requirements.

### EVECFG—%INCLUDE Statement for CICS Automation Control File Entries

```
***** 00000100
* (C) COPYRIGHT IBM CORP. 1990, 1995 - ALL RIGHTS RESERVED * 00
* * 00000300
* LICENSED MATERIALS - PROPERTY OF IBM * 00000400
* RESTRICTED MATERIALS OF IBM * 00000500
* 5685-151 * 00000600
* * 00000700
* US GOVERNMENT USERS RESTRICTED RIGHTS - * 00000800
* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY * 00000900
* GSA ADP SCHEDULE CONTRACT WITH IBM CORP. * 00001000
***** 00001100
* APAR# DATE * 00001200
* ----- * 00002000
* * 00002100
***** 00003000
%INCLUDE EVECFG01 00010000
```

### EVECFG01—Sample CICS Automation Control File Entries

```

* (C) COPYRIGHT IBM CORP. 1990,1994 - ALL RIGHTS RESERVED *
* *
* LICENSED MATERIALS - PROPERTY OF IBM *
* RESTRICTED MATERIALS OF IBM *
* 5685-151 *
* *
* US GOVERNMENT USERS RESTRICTED RIGHTS - *
* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY *
* GSA ADP SCHEDULE CONTRACT WITH IBM CORP. *

* APAR# DATE *
* ----- *
* $09=OW35605,V1R4,30Nov98,APC(JK): CICS Wave 2. *
* $08=OW32834,V1R4,11Jun98,APC(NS): Cater for CICS TS V1R2 changes. *
* $07=OW30931,V1R4,29Dec97,APC(AVF): Add EVEEATHR to resident list. *
* $06=OW28232,V1R4,22Sep97,APC(RW): Add RESIDENT EVEEYGRS & EVEERTRN *
* $05=OW08599 12/16/94 JS AOC CICS DOES NOT ISSUE STORAGE VIOLATION *
* COMMANDS CODED IN THE CONTROL FILE *
* *
* $04=PN57888 06/07/94 SW SAMPLE MESSAGE TABLE EVEMSG01 IS INCORRECT *
* MSGDFHSI1578 AND MSGDFHSI1579 SHOULD BE *
* MSGDFHSI1578D AND MSGMSGSI1579D, *
* *
* $03=PN52486 05/13/94 SW CICS AO SUPPORT FOR CICS V4 *
* *
* $02=PN45133 08/31/93 JS CICS AO CHANGES REQUIRED FOR COMPATIBILITY *
* WITH AOC/MVS V1R2 568515100 *
* *
* $01=PN42043 06/14/93 JS AFTER OY51838/UY91600 AOCORESTART COMMAND *
* NEEDS SUBSYS AS A PARM *
* *
```

```

* PN31479 11/25/92 JS CICS AO MESSAGE TABLE (EVMSG01) AND *
* ABCODESYSTM ENTRY FOR MSGDFH0605 INCORRECT,*
* ALSO U303 IN ENTRY FOR DFH0606 *
*
* PN31114 11/11/92 JS USE &EHKVAR1 IN RCVRTRAN STATEMENTS *
* PN15140 02/13/92 NS CICS AO SAMPLE EVECFG01 ENTRY FOR PPI AUTOOP*
* CICS PPI DOES NOT MATCH DOCUMENTATION WHERE *
* IT IS CALLED CICS CPPI *
*
* IW00002 04/29/91 Add missing commas to RCVRVIOL and ABCODTRAN *
* IW00088 06/12/91 Change health check sample *
* IW00094 06/12/91 Remove CRITMSG codes entry *
* PN02102 07/22/91 Correct Connection DAY= entry *

*
* AOC/CICS CONTROL FILE DEFINITIONS *
*
* PART 1 - DEFINITIONS WHICH ARE SUPPORTED THOUGH ISPF DIALOGS *

* AUTOOPS ENTRIES - DEFINE THE AUTOMATED OPERATORS *
*
* NOTE: THESE ENTRIES SHOULD BE DEFINED USING AOC/MVS ISPF DIALOGS *

*
* CICS MSTR - ASSIGNED ALL CICS MESSAGES
*
* AUTOOPS CICS MSTR, ID=AUTCICS, MSG=(EVE*, DFH*, AND*)
*
* CICS CPPI - HANDLES DEBLOCKING PPI INPUT
* NOTE: THIS GLOBAL NAME (CICS CPPI) IS NEVER USED IN ANY CICS AO
* ROUTINE. IT IS USED ONLY TO GET THE AUTOTASK AUTCPPI
* STARTED. WHATEVER IS CODED ON THE ID= OPERAND MUST MATCH
* WHAT IS CODED IN THE EVENTASK INITIALIZATION MEMBER ON THE
* SERVER STATEMENTS. IN THIS EXAMPLE, AUTCPPI IS SPECIFIED
* IN THE EVENTASK MEMBER.
*
* AUTOOPS CICS CPPI, ID=AUTCPPI
*
* CICS OPxx - HANDLES PROCESSING FOR 1 OR MORE CICS's
* SPECIFIED ON CICSNTL ENTRY, AUTOOPS=
* MULTIPLE CICS OPxx ENTRIES MAY BE SPECIFIED
* TWO AUTOTASKS ARE RECOMMENDED FOR EACH CICS OPxx
*
* AUTOOPS CICS OP01, ID=(AUTC01A, AUTC01B)
*

* ENVIRON ENTRIES - SYSTEM-WIDE ENVIRONMENT DEFINITIONS *
*
* NOTE: THESE ENTRIES SHOULD BE DEFINED USING AOC/MVS ISPF DIALOGS *

*
* ENVIRON SETUP, EXITS=(EVEEIIINT) Add initialization exits
*
* ENVIRON TIMEOUT, CEMT=29 Add CEMT time parameter
*

* APPLICATION ENTRIES - DEFINITIONS FOR EACH SEPARATE CICS SYSTEM *
*
* NOTE: THESE ENTRIES SHOULD BE DEFINED USING AOC/MVS ISPF DIALOGS *

*

```

```

*SUBSYSTEM CICST1,
* STARTCMD=YES, CUSTOMER STARTUP COMMAND REQUIRED
* JOB=CICST1, (other AOC/MVS specific
* PARENT=VTAM, parameters can be specified)
* SHUTDLY=00:03:00
*
*INITSTART CICST1, INITSTART EXIT REQUIRED BY AOC/CICS
* AUTO=E,
* EXITS=(EVEEIEXT)
*
*RESTART CICST1, RESTART EXIT REQUIRED BY AOC/CICS
* AUTO=E,
* EXITS=(EVEEIEXT)
*
*
*MINOR RESOURCE FLAGS FOR AOC/CICS
*RECOVERY CICST1.TRAN,AUTO=Y TRANSACTION RECOVERY
*RECOVERY CICST1.SOS,AUTO=Y SHORT-ON-STORAGE RECOVERY
*RECOVERY CICST1.VIOL,AUTO=Y STORAGE VIOLATIONS RECOVERY
*RECOVERY CICST1.AUTO,AUTO=Y AUTOINSTALL FAILURE ACTIONS
*RECOVERY CICST1.VTAMACB,AUTO=Y VTAM ACB RECOVERY
*

* ENTRIES FOR STARTUP, SHUTDOWN AND MSG RESPONSES *
*
* NOTE: THESE ENTRIES SHOULD BE DEFINED USING AOC/MVS ISPF DIALOGS *

*CICST1 STARTUP,CMD='MVS S CICST1' STARTUP COMMAND(S)
*
*CICST1 ACoreSTART,CMD=(,,'CICSRSYC CICST1') RESYNCH COMMAND
*
*CICST1 DFH1578,REPLY='CANCEL' PLTPI NOT FOUND
*
*CICST1 DFHSI1578D,REPLY='CANCEL' PLTPI NOT FOUND
*
*CICST1 DFH1579,REPLY='CANCEL' PLTPI PROGRAM NOT FND
*
*CICST1 DFHSI1579D,REPLY='CANCEL' PLTPI PROGRAM NOT FND
*
*CICST1 DFHSI1580D, PLTPI PROGRAM ABEND @08A
* REPLY=(,,'CANCEL')
*
*CICST1 DFHKE0408D, ARM REG FAILURE 3@09A
* REPLY=(ARMSTART,, 'AUTO'),
* REPLY=(NOARMSTART,, 'ASIS')
*
*CICST1 DFH1596,REPLY='CANCEL' NUCLEUS PROG NOT FND
*
*CICST1 DFH1307,REPLY='CANCEL' INTVL CONTROL FAILED
*
*CICST1 DFH1308,REPLY='CANCEL' TEMP STORAGE FAILURE
*
*CICST1 DFHTS1308,REPLY='CANCEL' TEMP STORAGE FAILURE
*
*CICST1 UP,CMD='MSG LOG,PERFORM UP PROCESSING' CICS IS UP
*
*CICST1 PPIACTIVE,CMD='MSG LOG,PERFORM PPI UP PROC' PPI COMMUNICATION
*
*CICST1 SHUTNORM, NORMAL SHUTDOWN
* CMD=(PASS1,, 'CICSSHUT NORMAL'),
* CMD=(PASS2,, 'CICSPURG'),
* CMD=(PASS3,, 'MVS C CICST1')
*
*CICST1 SHUTIMMED, IMMEDIATE SHUTDOWN
* CMD=(PASS1,, 'CICSSHUT IMMEDIATE'),

```

```

* CMD=(PASS2,, 'MVS C CICST1')
*
*CICST1 SHUTFORCE, FORCE SHUTDOWN
* CMD=(PASS1,, 'MVS C CICST1')
*
*CICST1 AUTODOWN, CMD='MSG LOG,PERFORM DOWN PROCESSING' CICS DOWN
*CICST1 STOPPED, CMD='MSG LOG,PERFORM DOWN PROCESSING' CICS DOWN
*CICST1 CTLDOWN, CMD='MSG LOG,PERFORM DOWN PROCESSING' CICS DOWN
*CICST1 ABENDED, CMD='MSG LOG,PERFORM DOWN PROCESSING' CICS DOWN
*

* Signon example. Necessary when autotasks must sign on to *
* CICS before issuing commands (CICS 2.1.2 level) *
* *
* Restart and shutdown commands are routed to the defined *
* automated operator (CICSOP01 in this example) *
* *
* AUTOTASK names and passwords are defined in EVEEX081 *

*
*CICST1 UP, CMD='EVEEX080 CICST1', SIGNON AUTOTASK
* CMD='MSG LOG,PERFORM UP PROCESSING' CICS IS UP
*
*CICST1 ACORESTART, CMD=(,CICSOP01,'EVEEX080 CICST1'), SIGNON
* CMD=(,,'CICRSYC CICST1') RESYNCH
*
*CICST1 SHUTNORM, NORMAL SHUTDOWN
* CMD=(PASS1,CICSOP01,'CICSSHUT NORMAL CICST1'),
* CMD=(PASS2,CICSOP01,'CICSPURG CICST1'),
* CMD=(PASS3,, 'MVS C CICST1')
*
*CICST1 SHUTIMMED, IMMEDIATE SHUTDOWN
* CMD=(PASS1,CICSOP01,'CICSSHUT IMMEDIATE CICST1'),
* CMD=(PASS2,, 'MVS C CICST1')
*

* RESIDENT CLISTS - DEFINED ONCE FOR EACH AOC/MVS SYSTEM *
* *
* NOTE: THESE ENTRIES SHOULD BE DEFINED USING AOC/MVS ISPF DIALOGS *

*RESIDENT EVEEX017
*RESIDENT EVEEMSG
*RESIDENT EVEEMSG1
*RESIDENT EVEECVR
*RESIDENT EVEECVR1
*RESIDENT EVEE0000
*RESIDENT EVEEIEXT
*RESIDENT EVEETEXT
*RESIDENT EVEESCHK
*RESIDENT EVEED001
*RESIDENT EVEEX100
*
* 2@06A
*RESIDENT EVEEYGRS
*RESIDENT EVEERTRN
*
* 1@07A
*RESIDENT EVEEATHR
*

* SDF COLOR DEFINITION ENTRIES - DEFINED ONCE PER AOC/MVS SYSTEM *
* *
* NOTE: THESE ENTRIES SHOULD BE DEFINED USING AOC/MVS ISPF DIALOGS *

*SDF CICSLMT,PR=200,CO=R,CLEAR=(Y,RV) LINK MONITORING
*SDF CICSHLT,PR=200,CO=R,CLEAR=(Y,RV) HEALTH CHECKING
*SDF CICSTRAN,PR=200,CO=R,CLEAR=(Y,RV) TRANSACTION ABENDS

```



```

*CPSMCICS12 ABENDED,
*CMD=(,,'MSG LOG,PERFORM DOWN PROCESSING')
*
* MESSAGE TABLE FOR : CPSMCICS12 ACORESTART
*
*CPSMCICS12 ACORESTART,
*CMD=(,,'CICSRSYC &SUBSJOB')
*
* MESSAGE TABLE FOR : CPSMCICS12 AUTODOWN
*
*CPSMCICS12 AUTODOWN,
*CMD=(,,'MSG LOG,PERFORM DOWN PROCESSING')
*
* MESSAGE TABLE FOR : CPSMCICS12 CTLDOWN
*
*CPSMCICS12 CTLDOWN,
*CMD=(,,'MSG LOG,PERFORM DOWN PROCESSING')
*
* MESSAGE TABLE FOR : CPSMCICS12 DFHKE0408D
*
*CPSMCICS12 DFHKE0408D,
*REPLY=(ARMSTART,, 'AUTO'),
*REPLY=(NOARMSTART,, 'ASIS')
*
* MESSAGE TABLE FOR : CPSMCICS12 DFHSI1578D
*
*CPSMCICS12 DFHSI1578D,
*REPLY=(,,'CANCEL')
*
* MESSAGE TABLE FOR : CPSMCICS12 DFHSI1579D
*
*CPSMCICS12 DFHSI1579D,
*REPLY=(,,'CANCEL')
*
* MESSAGE TABLE FOR : CPSMCICS12 DFHSI1580D
*
*CPSMCICS12 DFHSI1580D,
*REPLY=(,,'CANCEL')
*
* MESSAGE TABLE FOR : CPSMCICS12 DFHTS1308
*
*CPSMCICS12 DFHTS1308,
*REPLY=(,,'CANCEL')
*
* MESSAGE TABLE FOR : CPSMCICS12 PPIACTIVE
*
*CPSMCICS12 PPIACTIVE,
*CMD=(,,'MSG LOG,PERFORM PPI UP PROC')
*
* MESSAGE TABLE FOR : CPSMCICS12 SHUTFORCE
*
*CPSMCICS12 SHUTFORCE,
*CMD=(PASS1,, 'MVS C &SUBSJOB')
*
* MESSAGE TABLE FOR : CPSMCICS12 SHUTIMMED
*
*CPSMCICS12 SHUTIMMED,
*CMD=(PASS1,, 'CICSSHUT IMMEDIATE'),
*CMD=(PASS2,, 'MVS C &SUBSJOB')
*
* MESSAGE TABLE FOR : CPSMCICS12 SHUTNORM
*
*CPSMCICS12 SHUTNORM,
*CMD=(PASS1,, 'CICSSHUT NORMAL'),

```

```

*CMD=(PASS2,, 'CICSPURG'),
*CMD=(PASS3,, 'MVS C &SUBSJOB')
*
* MESSAGE TABLE FOR : CPSMCICS12 STARTUP
*
*CPSMCICS12 STARTUP,
*CMD=(,, 'MVS S EYUMAS, JOBNAME=&SUBSJOB, CICS=&SUBSJOB')
*
* MESSAGE TABLE FOR : CPSMCICS12 STOPPED
*
*CPSMCICS12 STOPPED,
*CMD=(,, 'MSG LOG, PERFORM DOWN PROCESSING')
*
* MESSAGE TABLE FOR : CPSMCICS12 UP
*
*CPSMCICS12 UP,
*CMD=(,, 'MSG LOG, PERFORM UP PROCESSING')
*
* NO SECONDARY ASSOCIATIONS FOR APPLICATION : CPSM_CICSTS12_CLASS
*
*
* End of Run at 98/11/19 03:36:32
*-----
* Title : Application Details Policy
* Desc : Standard CMAS TS V1.2 Class definition
*-----
*
*SUBSYSTEM_CLASS CPSMCMAS12,
* PARENT=(VTAM),
* SHUTDLY=00:03:00,
* SDESC='Standard CMAS TS V1.2 Class definition'
*
* NO THRESHOLDS DEFINED FOR : CPSM_CMASTS12_CLASS
*
*
* FLAG : INITSTART SETTINGS FOR : CPSMCMAS12
*
*INITSTART CPSMCMAS12,
* AUTO=E,
* EXITS=(EVEEEXT)
*
* FLAG : RESTART SETTINGS FOR : CPSMCMAS12
*
*RESTART CPSMCMAS12,
* AUTO=E,
* EXITS=(EVEEEXT)
*
* FLAG : RECOVERY SETTINGS FOR : CPSMCMAS12.AUTO
*
*RECOVERY CPSMCMAS12.AUTO,
* AUTO=Y
*
* FLAG : RECOVERY SETTINGS FOR : CPSMCMAS12.SOS
*
*RECOVERY CPSMCMAS12.SOS,
* AUTO=Y
*
* FLAG : RECOVERY SETTINGS FOR : CPSMCMAS12.TRAN
*
*RECOVERY CPSMCMAS12.TRAN,
* AUTO=Y
*
* FLAG : RECOVERY SETTINGS FOR : CPSMCMAS12.VIOL
*

```

```

*RECOVERY CPSMCMA12.VIOL,
* AUTO=Y
*
* FLAG : RECOVERY SETTINGS FOR : CPSMCMA12.VTAMACB
*
*RECOVERY CPSMCMA12.VTAMACB,
* AUTO=Y
*
* MESSAGE TABLE FOR : CPSMCMA12 ACORESTART
*
*CPSMCMA12 ACORESTART,
*CMD=(,,'CICSRSYC &SUBSJOB')
*
* MESSAGE TABLE FOR : CPSMCMA12 DFHKE0408D
*
*CPSMCMA12 DFHKE0408D,
*REPLY=(ARMSTART,, 'AUTO'),
*REPLY=(NOARMSTART,, 'ASIS')
*
* MESSAGE TABLE FOR : CPSMCMA12 DFHSI1578D
*
*CPSMCMA12 DFHSI1578D,
*REPLY=(,,'CANCEL')
*
* MESSAGE TABLE FOR : CPSMCMA12 DFHSI1579D
*
*CPSMCMA12 DFHSI1579D,
*REPLY=(,,'CANCEL')
*
* MESSAGE TABLE FOR : CPSMCMA12 DFHSI1580D
*
*CPSMCMA12 DFHSI1580D,
*REPLY=(,,'CANCEL')
*
* MESSAGE TABLE FOR : CPSMCMA12 DFHTS1308
*
*CPSMCMA12 DFHTS1308,
*REPLY=(,,'CANCEL')
*
* MESSAGE TABLE FOR : CPSMCMA12 SHUTFORCE
*
*CPSMCMA12 SHUTFORCE,
*CMD=(PASS1,, 'MVS C &SUBSJOB')
*
* MESSAGE TABLE FOR : CPSMCMA12 SHUTIMMED
*
*CPSMCMA12 SHUTIMMED,
*CMD=(PASS1,, 'EVEECMSH'),
*CMD=(PASS2,, 'CICSSHUT IMMEDIATE'),
*CMD=(PASS3,, 'MVS C &SUBSJOB')
*
* MESSAGE TABLE FOR : CPSMCMA12 SHUTNORM
*
*CPSMCMA12 SHUTNORM,
*CMD=(PASS1,, 'EVEECMSH'),
*CMD=(PASS2,, 'CICSSHUT NORMAL'),
*CMD=(PASS3,, 'CICSPURG'),
*CMD=(PASS4,, 'MVS C &SUBSJOB')
*
* NO SECONDARY ASSOCIATIONS FOR APPLICATION : CPSM_CMASTS12_CLASS
*
*
* End of Run at 98/11/19 03:36:38
*-----

```



```

* Title : Application Details Policy
* Desc : Sample CICSplex CICS AOR
*-----
*
*SUBSYSTEM EYUMAS1A,
* PARENT=(VTAM,EYUCMS1A),
* STARTCMD=YES,
* PROCNAME=EYUMAS,
* ARMNAME=SYSCICS_IPSAMC1A,
* OBJECTCLASS=CPSMCICS12,
* JOB=EYUMAS1A,
* SDESC='CICSplex KEY1 CICS AOR'
*
* NO THRESHOLDS DEFINED FOR : EYUMAS1A
*
*
* NO AUTOMATION FLAG DETAILS SPECIFIED FOR APPLICATION : EYUMAS1A
*
* MESSAGE TABLE FOR : EYUMAS1A STARTUP
*
*EYUMAS1A STARTUP,
*CMD=(,,'MVS S EYUMAS,JOBNAME=EYUMAS1A,CICS=EYUMAS1A')
*
* NO SECONDARY ASSOCIATIONS FOR APPLICATION : EYUMAS1A
*
*
* End of Run at 98/11/17 02:26:25
*-----
* Title : Application Details Policy
* Desc : Sample CICSplex CICS Manager Address Space (CMAS)
*-----
*
*SUBSYSTEM EYUCMS1A,
* PARENT=(VTAM),
* STARTCMD=YES,
* PROCNAME=EYUCMAS,
* ARMNAME=SYSCICS_IPSAMCM1,
* OBJECTCLASS=CPSMCMAS12,
* JOB=EYUCMS1A,
* SDESC='CICSplex KEY1 CICS Manager Address Space'
*
* NO THRESHOLDS DEFINED FOR : EYUCMS1A
*
*
* NO AUTOMATION FLAG DETAILS SPECIFIED FOR APPLICATION : EYUCMS1A
*
* MESSAGE TABLE FOR : EYUCMS1A STARTUP
*
*EYUCMS1A STARTUP,
*CMD=(,,'MVS S EYUCMAS,JOBNAME=EYUCMS1A,CMAS=EYUCMS1A')
*
* NO SECONDARY ASSOCIATIONS FOR APPLICATION : EYUCMS1A
*
*
* End of Run at 98/11/17 02:26:21
*-----
* Title : Application Details Policy
* Desc : CICSplex Control Address Space (CAS)
*-----
*
*SUBSYSTEM EYUCAS1A,
* PARENT=(VTAM),

```

```

* JOBTYP= MVS,
* SHUTDLY=00:02:00,
* STRTDLY=00:02:00,
* TERMDLY=00:02:00,
* PROCNAME=EYUCAS,
* JOB=EYUCAS1A,
* SDESC='CICSplex KEY1 Control Address Space'
*
* NO THRESHOLDS DEFINED FOR : EYUCAS1A
*
*
* FLAG : AUTOMATION SETTINGS FOR : EYUCAS1A
*
*AUTOMATION EYUCAS1A,
* AUTO=Y
*
* FLAG : RECOVERY SETTINGS FOR : EYUCAS1A
*
*RECOVERY EYUCAS1A,
* AUTO=Y
*
* FLAG : START SETTINGS FOR : EYUCAS1A
*
*START EYUCAS1A,
* AUTO=Y
*
* FLAG : TERMINATE SETTINGS FOR : EYUCAS1A
*
*TERMINATE EYUCAS1A,
* AUTO=Y
*
* FLAG : INITSTART SETTINGS FOR : EYUCAS1A
*
*INITSTART EYUCAS1A,
* AUTO=Y
*
* FLAG : RESTART SETTINGS FOR : EYUCAS1A
*
*RESTART EYUCAS1A,
* AUTO=Y
*
* MESSAGE TABLE FOR : EYUCAS1A SHUTFORCE
*
*EYUCAS1A SHUTFORCE,
*CMD=(PASS1,, 'MVS C &SUBSJOB')
*
* MESSAGE TABLE FOR : EYUCAS1A SHUTIMMED
*
*EYUCAS1A SHUTIMMED,
*CMD=(PASS1,, 'MVS C &SUBSJOB')
*
* MESSAGE TABLE FOR : EYUCAS1A SHUTNORM
*
*EYUCAS1A SHUTNORM,
*CMD=(PASS1,, 'MVS P &SUBSJOB'),
*CMD=(PASS2,, 'MVS C &SUBSJOB')
*
* NO SECONDARY ASSOCIATIONS FOR APPLICATION : EYUCAS1A
*
*
* End of Run at 98/11/17 02:26:17

*

```

```

* AOC/CICS CONTROL FILE DEFINITIONS *
*
* PART 2 - DEFINITIONS WHICH ARE NOT SUPPORTED THOUGH ISPF DIALOGS *

-----*
* APPLICATION ENTRIES - DEFINITIONS CODED FOR EACH SEPARATE CICS *
*
* NOTE: MOST OF THESE ENTRIES MUST BE CODED WITH AN EDITOR *

*
*
CICSCNTL CICST1, DEFINES A CICS FOR AOC/CICS
 AOC/MVS SUBSYSTEM NAME
 APPLID=(CICST1U,CICST1S), VTAM APPLID (generic,specific)
 MAJNODE=CICST1M, VTAM MAJOR NODE NAME
 AUTOOPS=CICSOP01, AUTOMATED OPERATOR
 KEYPOINTREQ=YES, KEYPOINT AT SHUTDOWN IS REQUIRED
 SMFID=215 SMF ID FOR AVAILABILITY RECORD
*
HEALTHCHK CICST1, HEALTH CHECKING
 FUNCTION=(0,EVECHLTH,00:10:00,30,AUTO,'SAMPLE COBOL HEALTH')
*
SERVICE CICST1, SERVICE PERIODS
 DAY=(DAILY,0745-1715),
 DAY=(WEEKEND,DOWN-DOWN)
*
TRIGGER CICST1, TRIGGER DEFINITION
 STARTUP=(SERVICE,JOB1DONE),
 SHUTDOWN=DOWN1,
 SHUTDOWN=(TEST1,TEST2,TEST3,TEST4,TEST5,TEST6)
EXTCOND JOB1DONE, TRIGGER EXPLANATIONS
 DESC='Job 1 Must be complete'
*
CONNECTION CICST1.C0T2, LINK MONITORING
 TYPE=CICS,
 DESC='FROM CICST1 TO CICST2',
 DAY=(DAILY,0900-1400),
 INTERVAL=27:00
*
LISTSHUT CICST1, LIST FOR CICSPURG
 INCLUDE=CSFE,
 INCLUDE=CEMT
*
 MINOR RESOURCE THRESHOLDS
THRESHOLDS CICST1.TRAN,CRIT=(2,00:05),FREQ=(4,01:00),INFR=(5,24:00)
THRESHOLDS CICST1.VIOL,CRIT=(3,00:05),FREQ=(3,01:00),INFR=(6,24:00)
THRESHOLDS CICST1.SOS,CRIT=(4,00:05),FREQ=(4,01:00),INFR=(7,24:00)
*
RCVRTRAN CICST1.TRAN, TRANSACTION ABEND RECOVERY
 CMD=(,,'MSG LOG,TRAN &EHKVAR1 FAILED')
*
RCVRTRAN CICST1.TRAN.ABCD, TRANSACTION ABEND RECOVERY
 CMD=(,,'MSG LOG,SPECIFIC TRAN &EHKVAR1 FAILED')
*
RCVRSOS CICST1.SOS, SHORT-ON-STORAGE RECOVERY
 TIMELIMIT=01:00,
 CMD=(,,'EVEERDMP CICST1')
*
* MVS modify command requires a comma @05C *
*
RCVRVIOL CICST1.VIOL, STORAGE VIOLATION RECOVERY
 CMD=(ON,, 'MVS F CICST1,CSFE DEBUG,FAQE=ON'),
 CMD=(OFF,, 'MVS F CICST1,CSFE DEBUG,FAQE=OFF')
*

```

```

* Note to CICS V3 users @05A *
* For CICS V3 the CSFE transaction has changed and requires
* different parameters as shown below. Please alter
* your RCVRVOL entries to use CHKSTSK rather than FAQE.
*
*RCVRVOL CICST1.VIOL,
* CMD=(ON,, 'MVS F CICST1,CSFE DEBUG,CHKSTSK=ALL'),
* CMD=(OFF,, 'MVS F CICST1,CSFE DEBUG,CHKSTSK=NONE')
*

*
RCVRAUTO CICST1.AUTO,TIMELIMIT=00:30 AUTOINSTALL ACTIONS
*
ABCODETRAN CICST1.TRAN, TRANSACTION ABEND CODES
 CODE=(CSFE,ATNI,*,EXCLUDE), TO INCLUDE OR EXCLUDE FROM
 CODE=(CSFE,AKC3,*,EXCLUDE), RECOVERY ACTIONS
 CODE=(*,ATNI,*,EXCLUDE),
 CODE=(*,*,*,INCLUDE)
*
ABCODESYSTM CICST1, SYSTEM ABEND CODES TO
 CODE=(IEF450I,S222,*,RESTART), ALLOW OR STOP RESTART
 CODE=(IEF450I,*,*,NORESTART), PROCESSING. UNRECOVERABLE
 CODE=(DFH0401,*,*,RESTART), ABENDS SHOULD BE NORESTART
 CODE=(DFHPC0401,*,*,RESTART),
 CODE=(DFH0404,*,*,RESTART),
 CODE=(DFHPC0404,*,*,RESTART),
 CODE=(DFH0409,*,*,RESTART),
 CODE=(DFHPC0409,*,*,RESTART),
 CODE=(DFH0501,*,*,RESTART),
 CODE=(DFH0504,*,*,RESTART),
 CODE=(DFH0601,*,*,RESTART),
 CODE=(DFHSR0601,*,*,RESTART),
 CODE=(DFH0602,*,*,RESTART),
 CODE=(DFHSR0602,*,*,RESTART),
 CODE=(DFH0603,*,*,RESTART),
 CODE=(DFHSR0603,*,*,RESTART),
 CODE=(DFH0607,*,*,RESTART),
 CODE=(DFH0612,*,*,RESTART),
 CODE=(DFHSR0612,*,*,RESTART),
 CODE=(DFH0613,*,*,RESTART),
 CODE=(DFHSR0613,*,*,RESTART),
 CODE=(DFH0614,*,*,RESTART),
 CODE=(DFHSR0614,*,*,RESTART),
 CODE=(DFH0615,*,*,RESTART),
 CODE=(DFHSR0615,*,*,RESTART),
 CODE=(DFH0619,*,*,RESTART),
 CODE=(DFH1797,*,*,RESTART),
 CODE=(DFHTM1797,*,*,RESTART),
 CODE=(DFH3784,*,*,RESTART),
 CODE=(DFH0408,*,*,NORESTART),
 CODE=(DFHPC0408,*,*,NORESTART),
 CODE=(DFH1306,*,*,NORESTART),
 CODE=(DFHTS1306,*,*,NORESTART),
 CODE=(DFH5725,*,*,NORESTART),
 CODE=(DFHER5725,*,*,NORESTART),
 CODE=(DFH0606,U305,*,NORESTART),
 CODE=(DFH0606,*,*,RESTART),
 CODE=(DFHSR0606,U305,*,NORESTART),
 CODE=(DFHSR0606,*,*,RESTART),
 CODE=(DFH1001I,*,*,NORESTART),
 CODE=(DFHTC1001I,*,*,NORESTART),
 CODE=(DFHAP0001,*,*,RESTART),
 CODE=(DFHSM0001,*,*,NORESTART)
*

```

156@09A

```

*-----
* Title : User Entry-Type Pairs
* Desc : CPSM MAS(CICS/TS AOR) ET Pairs
*-----
*
ABCODESYSTM EYUMAS1A,
CODE=(DFHSM0001,*,*,NORESTART),
CODE=(DFHAP0001,*,*,RESTART),
CODE=(DFHTC1001I,*,*,NORESTART),
CODE=(DFH1001I,*,*,NORESTART),
CODE=(DFHSR0606,*,*,RESTART),
CODE=(DFHSR0606,U305,*,NORESTART),
CODE=(DFH0606,*,*,RESTART),
CODE=(DFH0606,U305,*,NORESTART),
CODE=(DFHER5725,*,*,NORESTART),
CODE=(DFH5725,*,*,NORESTART),
CODE=(DFHTS1306,*,*,NORESTART),
CODE=(DFH1306,*,*,NORESTART),
CODE=(DFHPC0408,*,*,NORESTART),
CODE=(DFH0408,*,*,NORESTART),
CODE=(DFH3784,*,*,RESTART),
CODE=(DFHTM1797,*,*,RESTART),
CODE=(DFH1797,*,*,RESTART),
CODE=(DFH0619,*,*,RESTART),
CODE=(DFHSR0615,*,*,RESTART),
CODE=(DFH0615,*,*,RESTART),
CODE=(DFHSR0614,*,*,RESTART),
CODE=(DFH0614,*,*,RESTART),
CODE=(DFHSR0613,*,*,RESTART),
CODE=(DFH0613,*,*,RESTART),
CODE=(DFHSR0612,*,*,RESTART),
CODE=(DFH0612,*,*,RESTART),
CODE=(DFH0607,*,*,RESTART),
CODE=(DFHSR0603,*,*,RESTART),
CODE=(DFH0603,*,*,RESTART),
CODE=(DFHSR0602,*,*,RESTART),
CODE=(DFH0602,*,*,RESTART),
CODE=(DFHSR0601,*,*,RESTART),
CODE=(DFH0601,*,*,RESTART),
CODE=(DFH0504,*,*,RESTART),
CODE=(DFH0501,*,*,RESTART),
CODE=(DFHPC0409,*,*,RESTART),
CODE=(DFH0409,*,*,RESTART),
CODE=(DFHPC0404,*,*,RESTART),
CODE=(DFH0404,*,*,RESTART),
CODE=(DFHPC0401,*,*,RESTART),
CODE=(DFH0401,*,*,RESTART),
CODE=(IEF450I,*,*,NORESTART),
CODE=(IEF450I,S222,*,RESTART)
CICSCNTL EYUMAS1A,
SMFID=215,
KEYPOINTREQ=YES,
AUTOOPS=CICSOP01,
MAJNODE=MAJNODE,
APPLID=APPLID
LISTSHUT EYUMAS1A,
INCLUDE=CEMT,
INCLUDE=CSFE
RCVRAUTO EYUMAS1A.AUTO,
TIMELIMIT=00:30
RCVRSOS EYUMAS1A.SOS,
CMD=(,,'EVEERDMP &SUBSJOB'),
TIMELIMIT=01:00
RCVRVIOL EYUMAS1A.VIOL,

```

```

CMD=(OFF,, 'MVS F &SUBSJOB,CSFE DEBUG,CHKSTSK=NONE'),
CMD=(ON,, 'MVS F &SUBSJOB,CSFE DEBUG,CHKSTSK=ALL')
SERVICE EYUMAS1A,
DAY=(DAILY,0520-2300)
THRESHOLDS EYUMAS1A.SOS,
INFR=(7,24:00),
FREQ=(4,01:00),
CRIT=(4,00:05)
THRESHOLDS EYUMAS1A.VIOL,
INFR=(6,24:00),
FREQ=(3,01:00),
CRIT=(3,00:05)
TRIGGER EYUMAS1A,
SHUTDOWN=(SERVICE),
STARTUP=(SERVICE)
*
* End of Run at 98/11/17 02:27:59
*-----
* Title : User Entry-Type Pairs
* Desc : CPSM CMAS ET Pairs
*-----
*
ABCODESYSTM EYUCMS1A,
CODE=(DFHSM0001,*,*,NORESTART),
CODE=(DFHAP0001,*,*,RESTART),
CODE=(DFHTC1001I,*,*,NORESTART),
CODE=(DFH1001I,*,*,NORESTART),
CODE=(DFHSR0606,*,*,RESTART),
CODE=(DFHSR0606,U305,*,NORESTART),
CODE=(DFH0606,*,*,RESTART),
CODE=(DFH0606,U305,*,NORESTART),
CODE=(DFHER5725,*,*,NORESTART),
CODE=(DFH5725,*,*,NORESTART),
CODE=(DFHTS1306,*,*,NORESTART),
CODE=(DFH1306,*,*,NORESTART),
CODE=(DFHPC0408,*,*,NORESTART),
CODE=(DFH0408,*,*,NORESTART),
CODE=(DFH3784,*,*,RESTART),
CODE=(DFHTM1797,*,*,RESTART),
CODE=(DFH1797,*,*,RESTART),
CODE=(DFH0619,*,*,RESTART),
CODE=(DFHSR0615,*,*,RESTART),
CODE=(DFH0615,*,*,RESTART),
CODE=(DFHSR0614,*,*,RESTART),
CODE=(DFH0614,*,*,RESTART),
CODE=(DFHSR0613,*,*,RESTART),
CODE=(DFH0613,*,*,RESTART),
CODE=(DFHSR0612,*,*,RESTART),
CODE=(DFH0612,*,*,RESTART),
CODE=(DFH0607,*,*,RESTART),
CODE=(DFHSR0603,*,*,RESTART),
CODE=(DFH0603,*,*,RESTART),
CODE=(DFHSR0602,*,*,RESTART),
CODE=(DFH0602,*,*,RESTART),
CODE=(DFHSR0601,*,*,RESTART),
CODE=(DFH0601,*,*,RESTART),
CODE=(DFH0504,*,*,RESTART),
CODE=(DFH0501,*,*,RESTART),
CODE=(DFHPC0409,*,*,RESTART),
CODE=(DFH0409,*,*,RESTART),
CODE=(DFHPC0404,*,*,RESTART),
CODE=(DFH0404,*,*,RESTART),
CODE=(DFHPC0401,*,*,RESTART),
CODE=(DFH0401,*,*,RESTART),

```

```

CODE=(IEF450I,*,*,NORESTART),
CODE=(IEF450I,S222,*,RESTART)
CICSCTL EYUCMS1A,
SMFID=215,
KEYPOINTREQ=YES,
AUTOOPS=CICSOP01,
MAJNODE=MAJNODE,
APPLID=APPLID
LISTSHUT EYUCMS1A,
INCLUDE=CEMT,
INCLUDE=CSFE
RCVRAUTO EYUCMS1A.AUTO,
TIMELIMIT=00:30
RCVRSOS EYUCMS1A.SOS,
CMD=(,,'EVEERDMP &SUBSJOB'),
TIMELIMIT=01:00
RCVRVIOL EYUCMS1A.VIOL,
CMD=(OFF,,'MVS F &SUBSJOB,CSFE DEBUG,CHKSTSK=NONE'),
CMD=(ON,,'MVS F &SUBSJOB,CSFE DEBUG,CHKSTSK=ALL')
THRESHOLDS EYUCMS1A.SOS,
INFR=(7,24:00),
FREQ=(4,01:00),
CRIT=(4,00:05)
THRESHOLDS EYUCMS1A.VIOL,
INFR=(6,24:00),
FREQ=(3,01:00),
CRIT=(3,00:05)
*
* End of Run at 98/11/17 02:27:54

* STATE/ACTION TABLES - SPECIFIED ONCE IN THE CONTROL FILE *

* EACH CICS SHOULD BE SPECIFIED IN THE PRODUCT STATEMENT *

*
*
PRODUCT CICS,SUBSYS=(CICST1,AREA=CICS),
 SUBSYS=(CICST2,AREA=CICS),
 SUBSYS=(EYUCMS1A,AREA=CICS),
 SUBSYS=(EYUMAS1A,AREA=CICS)
*
AREA CICS,SOS=EVEESA01,
 VIOL=EVEESA02,
 AUTO=EVEESA03,
 VTAMACB=EVEESA04

```

@09C

## CICSMSG0 (EVEMSG00)—Early NetView Initialization Automation Table

```

***** 00001000
* (C) COPYRIGHT IBM CORP. 1990,1995 - ALL RIGHTS RESERVED
*
* LICENSED MATERIALS - PROPERTY OF IBM
* RESTRICTED MATERIALS OF IBM
* 5685-151
*
* US GOVERNMENT USERS RESTRICTED RIGHTS -
* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY
* GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
*
***** 00004000
***** 00005000
* APAR# DATE
* -----
* 1.3 04/25/94 JG CREATED FOR AOC 1.3
*
***** 00054000
*****00340IMS
* THESE ENTRIES ARE TO BE MERGED INTO DSIMSG00 OR DSITBL00 IN ORDER *00340IMS
* TO ENSURE PROPER OPERATION OF SINGLE POINT OF CONTROL FOR CICS *00340IMS
* AUTOMATION *00340IMS
*****00340IMS
* AT NETVIEW INITIALIZATION OR DOMAIN RESTART THERE MAY OCCUR A *00340IMS
* SITUATION WHEN DSIMSG/TBL00 IS STILL LOADED WHEN GATEWAYS ARE *00340IMS
* STARTED. THIS CAUSES MESSAGE TABLE TRAPS FOR SPOC TO BE MISSED *00340IMS
* AND SPOC NOT TO FUNCTION. THE ENTRIES BELOW PLACED IN DSIMSG/TBL00 *00340IMS
* WILL ENSURE THIS DOES NOT OCCUR. NOTE: SPOC IS SINGLE POINT OF *00340IMS
* CONTROL *00340IMS
*****00340IMS
----- 04800000
* INCLUDE FOR CICS MESSAGES * 04810000
----- 04820000
* 04830000
*%INCLUDE EVEMEVE0 04850000
* 04940000
----- 04800000
* INCLUDE FOR CONFLICT MESSAGES * 04810000
----- 04820000
* 04830000
*%INCLUDE EVEMCON0 04850000
* 04940000
***** 05630000
* END OF CICS AUTOMATION MESSAGE TABLE ENTRIES * 05640000
***** 05650000

```



## EVEMEVE0—EVE Messages (Included by CICSMSG0)

```

***** 00001000
* (C) COPYRIGHT IBM CORP. 1990,1995 - ALL RIGHTS RESERVED
*
* LICENSED MATERIALS - PROPERTY OF IBM
* RESTRICTED MATERIALS OF IBM
* 5685-151
*
* US GOVERNMENT USERS RESTRICTED RIGHTS -
* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY
* GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
*
***** 00004000
***** 00005000
* APAR# DATE * 00006000
* ----- -
* 1.3 04/25/94 JG CREATED FOR AOC 1.3 * 00007000
*
***** 00054000
* 04790000
* -----* 04800000
* CICS SINGLE POINT OF CONTROL SUPPORT MESSAGES * 04810000
* -----* 04820000
* 04830000
* CROSS-SYSTEM MESSAGE ROUTING 04840000
* IF MSGID = 'EVE697' 04850000
* THEN EXEC(CMD('EVESX003 ') ROUTE(ONE *)) 04860000
* DISPLAY(Y) NETLOG(Y); 04870000
* 04940000
* SINGLE POINT OF CONTROL 04940000
* 04940000
* IF MSGID = 'EVE710I' & TEXT = MESSAGE 04950000
* THEN EXEC(CMD('EVESXDMB 'MESSAGE) ROUTE(ONE *)) 04960000
* DISPLAY(Y) NETLOG(Y); 04970000
***** 05630000
* END OF CICS AUTOMATION EVEMEVE0 MESSAGE TABLE ENTRIES * 05640000
***** 05650000

```

## EVEMCON0—Messages in Conflict with AOC/MVS (Included by CICSMSG0)

```

***** 00001000
* (C) COPYRIGHT IBM CORP. 1994, 1995 - ALL RIGHTS RESERVED
*
* LICENSED MATERIALS - PROPERTY OF IBM
* RESTRICTED MATERIALS OF IBM
* 5685-151
*
* US GOVERNMENT USERS RESTRICTED RIGHTS -
* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY
* GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
*
***** 00004000
***** 00005000
* APAR# DATE * 00006000
* ----- ----- * 00007000
* 1.3 04/25/94 JG CREATED FOR AOC 1.3 *
*
***** 00054000
*****00340IMS
*
* WARNING ---- POSSIBLE CONFLICT WITH EXISTING MESSAGE TABLE STMTS 04340IMS
* WARNING ---- POSSIBLE CONFLICT WITH EXISTING MESSAGE TABLE STMTS 04350IMS
* WARNING ---- POSSIBLE CONFLICT WITH EXISTING MESSAGE TABLE STMTS 04350IMS
*
*
* WARNING ---- POSSIBLE CONFLICT WITH EXISTING MESSAGE TABLE STMTS 04350IMS
IF MSGID = 'AOF660I' & TEXT = MESSAGE & HDRMTYPE = 'E'
 THEN EXEC(CMD('EVEEX041 'MESSAGE) ROUTE(ONE *))
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(N) SYSLOG(N) CONTINUE(Y);
*
* WARNING ---- POSSIBLE CONFLICT WITH EXISTING MESSAGE TABLE STMTS 04340IMS
IF MSGID = 'AOF662I' & TEXT = MESSAGE & HDRMTYPE = 'E'
 THEN EXEC(CMD('EVEEX040 'MESSAGE) ROUTE(ONE *))
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(N) SYSLOG(N) CONTINUE(Y);
*
* NO FOCAL POINTS AVAILABLE * @01 * 03040IMS
* WARNING ----POSSIBLE CONFLICT WITH EXISTING MESSAGE TABLE STMTS 04380IMS
*
IF MSGID='AOF664I' & TEXT= MESSAGE & HDRMTYPE='E'
 THEN EXEC(CMD('EVEEX042 ' MESSAGE) ROUTE(ALL *))
 DISPLAY(Y) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(N) CONTINUE(Y);

```

## CICSMSG1 (EVEMSG01)—Production NetView Initialization Automation Table

```

***** 00001000
* (C) COPYRIGHT IBM CORP. 1990,1994,1998 - ALL RIGHTS RESERVED
*
* LICENSED MATERIALS - PROPERTY OF IBM
* RESTRICTED MATERIALS OF IBM
* 5685-151
*
* US GOVERNMENT USERS RESTRICTED RIGHTS -
* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY
* GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
*
***** 00004000
***** 00005000
* APAR# DATE * 00006000
* ----- -
* 1.3 04/25/94 JG CREATED FOR AOC 1.3 *
* $01=OW35605,V1R4,12OCT98,APC(JK): NEW FOR CICSplex SM *
* *
***** 00054000
***** 00071000
* INCLUDE FOR EVE* MESSAGES * 00360IMS
***** 00380IMS
* 00390IMS
*
* %INCLUDE EVEMEVE1
* 00390IMS
***** 00340IMS
* INCLUDE FOR DFH* MESSAGES * 00360IMS
***** 00380IMS
* 00390IMS
*
* %INCLUDE EVEMDFH1
* 00390IMS
***** 00340IMS
* INCLUDE FOR EYU* MESSAGES (CICSplex) @01A* 00360IMS
***** 00380IMS
* 00390IMS
*
* %INCLUDE EVEMEYU1
* 00390IMS
***** 00340IMS
* INCLUDE FOR CONFLICT MESSAGES * 00360IMS
***** 00380IMS
* 00390IMS
*
* %INCLUDE EVEMCON1
* 00390IMS
***** 00340IMS
* END OF CICS AUTOMATION MESSAGE TABLE ENTRIES * 00360IMS
***** 00380IMS

```

## EVEMEVE1—EVE Messages (Included by CICSMSG1)

```

***** 00001000
* (C) COPYRIGHT IBM CORP. 1990,1995 - ALL RIGHTS RESERVED
*
* LICENSED MATERIALS - PROPERTY OF IBM
* RESTRICTED MATERIALS OF IBM
* 5685-151
*
* US GOVERNMENT USERS RESTRICTED RIGHTS -
* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY
* GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
*
*
* 00003000
***** 00001000
***** 00005000
* APAR# DATE * 00006000
* ----- -
***** 00007000
***** 00001000
* Start of EVE message block * 00002000
* * 00003000
***** 00004000
* 00638000
*
* IF MSGID='EVE' .
* THEN
* BEGIN;
*
* 00219000
* CICS PPI NOW ACTIVE 00220000
* VTAMACB STATE ACTION - CICS PPI OPEN 00221000
* IF MSGID='EVE172I' 00222000
* THEN EXEC(CMD('EVESROUT EVEEI010 ') ROUTE(ALL *)) 00223000
* EXEC(CMD('EVESROUT EVEEY00S MSGID=PPIOPN') ROUTE(ALL *)) 00224000
* DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00225000
*
* 00480000
* MAJOR NODE RECOVERED MESSAGE 00493000
* IF MSGID ='EVE780I' & TOKEN(2)='APPL' & TOKEN(3)='SEGMENT' 00494000
* & TOKEN(4)=RESOURCE & TOKEN(5)='ACTIVE' 00495000
* THEN EXEC(CMD('EVEES411 ' RESOURCE)); 00496000
*
* 00497000
* APPLID RECOVERED MESSAGE 00498000
* IF MSGID ='EVE780I' & TOKEN(2)='APPL' 00499000
* & TOKEN(3)=RESOURCE & TOKEN(4)='ACTIVE' 00500000
* THEN EXEC(CMD('EVEES412 ' RESOURCE)); 00501000
* -----* 00502000
* CICS LINK RECOVERY - CDRM FAILURE * 00503000
* -----* 00504000
*
* 00505000
* CDRM RECOVERED MESSAGE 00506000
* IF MSGID ='EVE780I' & TOKEN(2)='CDRM' & TOKEN(3)=RESOURCE 00507000
* & TOKEN(4)='ACTIVE' 00508000
* THEN EXEC(CMD('EVEEZ001 ' RESOURCE)); 00509000
* -----* 00639000
* CICS SINGLE POINT OF CONTROL SUPPORT MESSAGES * 00640000
* -----* 00641000
*
* 00642000
* CROSS-SYSTEM MESSAGE ROUTING 00643000
* IF MSGID = 'EVE697' 00644000
* THEN EXEC(CMD('EVESX003 ') ROUTE(ONE *)) 00645000
* DISPLAY(Y) NETLOG(N); 00646000
*
* 00647000
* BROADCAST MESSAGES ACROSS SUBSYSTEMS 0648000
* IF MSGID = 'EVE755' 00649000
* THEN EXEC(CMD('EVEEB101 ') ROUTE(ONE *)) 00650000

```

|                                                                   |            |
|-------------------------------------------------------------------|------------|
| DISPLAY(Y) NETLOG(N);                                             | 00651000   |
| *                                                                 | 00652000   |
| * SINGLE POINT OF CONTROL                                         | 653000     |
| IF MSGID = 'EVE710I' & TEXT = MESSAGE                             | 00654000   |
| THEN EXEC( CMD('EVESXDMB 'MESSAGE) ROUTE(ONE *)))                 | 00655000   |
| DISPLAY(Y) NETLOG(N);                                             | 00656000   |
| *                                                                 |            |
| * NEW MESSAGE TO IDENTIFY FP OF CONTENTS OF ALL SUBSYSTEM MESSAGE | 657000     |
| IF MSGID = 'EVE715I' & TEXT = MESSAGE                             |            |
| THEN EXEC( CMD('EVEEAAST FP ' MESSAGE) ROUTE(ONE *)))             |            |
| DISPLAY (Y) NETLOG(N);                                            |            |
|                                                                   |            |
| * -----*                                                          | p0658000   |
| * CICS STARTUP CONTROL                                            | * 00659000 |
| * -----*                                                          | 00660000   |
| *                                                                 | 00669000   |
| * LINK MONITORING DF UPDATE                                       | 00670000   |
| *                                                                 | 00671000   |
| IF MSGID='EVE811I' & TEXT = MESSAGE                               | 00672000   |
| THEN EXEC(CMD('EVESROUT EVEED004 'MESSAGE) ROUTE(ALL *));         | 00673000   |
| *                                                                 | 00674000   |
| IF MSGID='EVE812I' & TEXT = MESSAGE                               | 00675000   |
| THEN EXEC(CMD('EVESROUT EVEED004 'MESSAGE) ROUTE(ALL *));         | 00676000   |
| *                                                                 | 00677000   |
| IF MSGID='EVE813I' & TEXT = MESSAGE                               | 00678000   |
| THEN EXEC(CMD('EVESROUT EVEED004 'MESSAGE) ROUTE(ALL *));         | 00679000   |
| *                                                                 | 00680000   |
| IF MSGID='EVE814I' & TEXT = MESSAGE                               | 00681000   |
| THEN EXEC(CMD('EVESROUT EVEED004 'MESSAGE) ROUTE(ALL *));         | 00682000   |
| *                                                                 | 00683000   |
| IF MSGID='EVE815I' & TEXT = MESSAGE                               | 00684000   |
| THEN EXEC(CMD('EVESROUT EVEED004 'MESSAGE) ROUTE(ALL *));         | 00685000   |
| *                                                                 | 00686000   |
| IF MSGID='EVE816I' & TEXT = MESSAGE                               | 00687000   |
| THEN EXEC(CMD('EVESROUT EVEED004 'MESSAGE) ROUTE(ALL *));         | 00688000   |
| *                                                                 | 00689000   |
| IF MSGID='EVE817I' & TEXT = MESSAGE                               | 00690000   |
| THEN EXEC(CMD('EVESROUT EVEED004 'MESSAGE) ROUTE(ALL *));         | 00691000   |
| *                                                                 | 00692000   |
| IF MSGID='EVE819I' & TEXT = MESSAGE                               | 00693000   |
| THEN EXEC(CMD('EVESROUT EVEED004 'MESSAGE) ROUTE(ALL *));         | 00694000   |
| *                                                                 | 00695000   |
| IF MSGID='EVE820I' & TEXT = MESSAGE                               | 00696000   |
| THEN EXEC(CMD('EVESROUT EVEED004 'MESSAGE) ROUTE(ALL *));         | 00697000   |
| *                                                                 | 00698000   |
| ALWAYS;                                                           |            |
| END;                                                              |            |

## EVEMDFH1—DFH Messages (Included by CICSMSG1)

```

* (C) COPYRIGHT IBM CORP. 1990,1995 - ALL RIGHTS RESERVED

*
* LICENSED MATERIALS - PROPERTY OF IBM
* 5685-151 (C) Copyright IBM Corp. 1990, 1995
*

* APAR# DATE
* -----
* $17=OW37400,V1R4,01Feb99,APC(ML): Unable to interpret msgDFHSI1500.*
* $16=OW35605,V1R4,20Oct98,APC(JK): CICS TS V1R1/R2 Wave 2 changes. *
* $15=OW32834,V1R4,22JU198,APC(NS): Cater for version 5 changes. *
* $14=OW33954,V1R4,20Ju198,APC(AE): Handle DFH1280 regardless whether*
* TOKEN(2) is a dash (-) or not. *
* $13=OW32121,V1R4,09Mar98,APC(RW): Trap unknown WTOR's *
* $12=OW31698,V1R4,23Feb98,APC(RW): Change DFH1280 to trap TOKEN(2) *
* $11=OW30723,V1R4,03Dec97,APC(RW): Uppercase msgDFHXG6513 text *
* $10=OW21476 07/15/96 HI MSGAOF206I ISSUED FOR MULTIPLE LINE CICS *
* MSGDFH1308. PROBLEM SAME AS APAR PN48038.*
*
* $11=OPER 09/30/95 LK Operator start, add call to EVEEAOPS *
* $09=OW10263 07/27/95 JS AOC CICS SETS ABENDING REGION TO STOPPING *
* STATUS, SHOULD BE "BREAKING". *
*
* $08=OW08596 12/13/94 JS AOC CICS MESSAGE TABLE TRAP FOR MSGDFS3463I*
* INCORRECT -- TRAPS WRONG MESSAGE *
*
* $07=PN58923 07/06/94 SW AFTER PTF UN63573, RECEIVES MSGEVE653E *
* FAILED TO LOCATE AREA FOR MSGDFHZC5966,*
* STATE ACTION TABLE NEEDS TO BE REVISED
*
* $06=PN57888 06/01/94 SW SAMPLE MESSAGE TABLE EVEMSG01 IS INCORRECT *
* MSGDFHSI1578 AND MSGDFHSI1579 SHOULD BE
* MSGDFHSI1578D AND MSGMSGSI1579D AND
* MSGDFHZC5966I SHOULD BE MSGDFHZC5966
*
* FEAT 1.3 05/09/94 JG Tailored for AOC 1.3
* $05=PN52486 03/11/94 MP CICS AO SUPPORT FOR CICS V4
* Added DFHSM0131, 132, 133 and 134
*
* 03/23/94 JK CICS AO SUPPORT FOR CICS V4
* Added DFHPA1918, DFHPC0401, 405, 408, 409,
* DFHTC1041, DFHRU2801, 2802, 2803,
* DFHIR3785, DFHER5725, DFHXG6400, 6401,
* 6405, 6409, 6500, 6501, 6513, and
* DFHTS1308
*
* $04=PN50608 01/21/94 JS MSGCNM516E VALUE SPECIFIED FOR MSGID
* FUNCTION IS TOO LONG WHEN LOADING MESSAGE
* TABLE FOR CICS AO (removing all +DFH).
*
* $03=PN48038 11/16/93 JS MSGAOF206I FOR NULL REPLYID WHEN MSGDFH1596*
* TRAPPED BY CICS AO AND IT HAS NOT BEEN
* ISSUED AS WTOR
*
* $02=PN45133 08/25/93 JS CICS AO CHANGES REQUIRED FOR COMPATIBILITY *
* WITH AOC/MVS V1R2 568515100
*
* $01=PN41690 06/07/93 MP MESSAGE IEE105I NUMBER CHANGED TO IEE115I *
* FOR MVS 4.3

```

|   |         |          |    |                                                 |            |
|---|---------|----------|----|-------------------------------------------------|------------|
| * |         |          |    |                                                 | * 00011000 |
| * | PN36057 | 02/27/93 | JS | CICSAO TRAP FOR SHORT-ON-STORAGE FAILS IN       | * 00011100 |
| * |         |          |    | CICS V3.2 AS MSGDFHSM0100 AND MSGDFHSM0101      | * 00011200 |
| * |         |          |    | DO NOT HAVE 'I' SUFFIX                          | * 00011300 |
| * |         |          |    |                                                 | * 00011400 |
| * | PN34918 | 02/12/93 | JS | CICSAO DOCUMENTS TAKEOVER STATUS BUT DOES       | * 00011500 |
| * |         |          |    | NOT SET IT                                      | * 00011600 |
| * |         |          |    |                                                 | * 00011700 |
| * | PN32138 | 01/18/93 | JS | ADD MSGDFH1592 TO EVEMSG01 FOR VTAM ACB         | * 00011800 |
| * |         |          |    | RECOVERY IN XRF ENVIRONMENT FOR BACKUP          | * 00011900 |
| * |         |          |    | SYSTEM                                          | * 00012000 |
| * |         |          |    |                                                 | * 00012100 |
| * | PN31479 | 11/25/92 | JS | CICSAO MESSAGE TABLE (EVEMSG01) AND             | * 00012200 |
| * |         |          |    | ABCODESYSTM ENTRY FOR MSGDFH0605 INCORRECT,     | * 00012300 |
| * |         |          |    | ALSO U303 IN ENTRY FOR DFH0606                  | * 00012400 |
| * |         |          |    |                                                 | * 00012500 |
| * | PN31114 | 11/16/92 | JS | MESSAGES FOR TRANSACTION RECOVERY ARE NOT       | * 00012600 |
| * |         |          |    | TRAPPED BECAUSE THEY HAVE AN 'I' SUFFIX.        | * 00019000 |
| * |         |          |    |                                                 | * 00019100 |
| * | PN30263 | 10/21/92 | RW | CICSAO MEMBER EVEMSG01 CONTAINS ENTRIES FOR     | * 00019200 |
| * |         |          |    | CICS ABEND MESSAGES THAT ARE RECOVERABLE        | * 00019300 |
| * |         |          |    | AND NOT TERMINAL TO CICS.                       | * 00019400 |
| * |         |          |    |                                                 | * 00019401 |
| * | PN17565 | 06/26/92 | MP | CICSAO SUPPORT FOR CICS V3R3                    | * 00019402 |
| * |         |          |    | ALL DFH06NN MESSAGES ARE NOW DFHSR06NN.         | * 00019403 |
| * |         |          |    | DFH1001I IS NOW DFHTC1001I. DFHTS1306,          | * 00019404 |
| * |         |          |    | DFHTS1307, DFHTS1310, DFHTS1311, DFHTS1313      | * 00019405 |
| * |         |          |    | DFHSI1500, DFHSI1502, DFHSI1517, DFHSI1572      | * 00019406 |
| * |         |          |    | DFHSI1578, DFHSI1579, DFHSI1596                 | * 00019407 |
| * |         |          |    | DFHTM1701, DFHTM1702, DFHTM1713, DFHTM1797      | * 00019408 |
| * |         |          |    | DFHSM0116, DFHSM0117                            | * 00019409 |
| * |         |          |    |                                                 | * 00019410 |
| * |         |          |    |                                                 | * 00019411 |
| * | PN12078 | 01/14/92 | TP | MESSAGES DFH2305 AND DFH2307 ARE NOT BEING      | * 00019412 |
| * |         |          |    | TRAPPED WHEN MAJOR NODE IS CLOSED.              | * 00019413 |
| * |         |          |    |                                                 | * 00020000 |
| * | PN12065 | 01/06/92 | TP | MESSAGE DFH1280 IS NOT PRODUCED ON ESA,         | * 00021000 |
| * |         |          |    | THEREFORE THE MESSAGE TRAP DOES NOT WORK ON     | * 00022000 |
| * |         |          |    | ESA.                                            | * 00023000 |
| * |         |          |    |                                                 | * 00024000 |
| * | PN11909 | 12/11/91 | TP | APPLICATION ABEND WITH MSG DFHAP0001 CAUSES     | * 00025000 |
| * |         |          |    | SUBSYSTEM TO BE PUT IN STOPPING STATUS.         | * 00026000 |
| * |         |          |    |                                                 | * 00027000 |
| * | PN07516 | 10/07/91 | TP | UNDER CICS V3.2.1 WHEN APPLID IS 8              | * 00028000 |
| * |         |          |    | CHARACTERS MSGDFHZC3463 IS ISSUED AS MLWTO      | * 00029000 |
| * |         |          |    | AND CICSAO PARSING IN MESSAGE TABLE FAILS       | * 00030000 |
| * |         |          |    |                                                 | * 00031000 |
| * | PN04063 | 08/20/91 | TP | MSGEVE601W STARTUP HUNG MSG IS MISLEADING       | * 00032000 |
| * |         |          |    | FOR CICS STANDBY SUBSYSTEM UNDER CICSAO         | * 00033000 |
| * |         |          |    | CONTROL.                                        | * 00034000 |
| * |         |          |    |                                                 | * 00035000 |
| * | PN04134 | 08/16/91 | TP | CICSAO DOES NOT UPDATE XRF STATUS BECAUSE       | * 00036000 |
| * |         |          |    | MSGDFH6509 USED IN MESSAGE TABLE (EVEMSG01)     | * 00037000 |
| * |         |          |    | DOES NOT EXIST                                  | * 00038000 |
| * |         |          |    |                                                 | * 00039000 |
| * | PN03535 | 08/13/91 | TP | THE SDF PANEL EVEDTRN1 (CICS TRANSACTION        | * 00040000 |
| * |         |          |    | MONITOR) SHOWED ERRORMSG AS THE THE MESSAGE     | * 00041000 |
| * |         |          |    | TEXT. THIS IS MEANINGLESS.                      | * 00042000 |
| * |         |          |    |                                                 | * 00043000 |
| * | PN02413 | 07/23/91 | MP | CICSAO MSGEVE144I EVESROUT UNSUCCESSFUL         | * 00044000 |
| * |         |          |    | RC=104 (020) FOR JOBANME INIT FOR MATCH ON      | * 00045000 |
| * |         |          |    | MSGIEF404I OR MSGIEF450I                        | * 00046000 |
| * |         |          |    |                                                 | * 00047000 |
| * | IW00033 | 05/28/91 |    | Correct handling of CTLDOWN and service periods | * 00048000 |

```

* IW00028 05/30/91 Change EVE780 msg for VTAM ACB recovery * 00049000
* IW00024 06/04/91 Change gateway msg trap from 546/548 to 660/662 * 00050000
* IW00067 06/04/91 Add route autcics for Sit replyid "and enter .." * 00051000
* IW00080 06/07/91 change msgid dfh6400, dfh6401, dfh6500, dfh6501 * 00052000
* IW00098 06/19/91 Trap AOF660 and AOF662 messages * 00053000
***** 00054000
***** 00054001
* * 00054002
* Start of DFH message block * 00054003
* * 00054004
***** 00054005
* 00638000
IF MSGID='DFH' . 00638100
THEN 00638200
BEGIN; 00638201
00638202
* TRAP CICS VERSION NUMBER 00638203
IF MSGID='DFH1500' 00638204
& TOKEN(4)='VERSION' & TOKEN(5)=VERNUM & TOKEN(6) = 'START-UP' 00638205
THEN EXEC(CMD('EVESROUT EVEEI004 'VERNUM) ROUTE(ALL *)) 00638206
DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638207
* 00638208
* TRAP CICS VERSION NUMBER @15C 00638209
IF MSGID='DFHSI1500' 00638210
& TOKEN(4)='Version' & TOKEN(5) = VERNUM 0063821
THEN EXEC(CMD('EVESROUT EVEEI004 'VERNUM) ROUTE(ALL *)) 00638212
DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638213
* 00638208
* TRAP CICS VERSION NUMBER V1.1 (V5.1) 4@17A 00638209
IF MSGID='DFHSI1500' 00638210
& TOKEN(6)='Version' & TOKEN(7) = VERNUM 0063821
THEN EXEC(CMD('EVESROUT EVEEI004 'VERNUM) ROUTE(ALL *)) 00638212
DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638213
* 00638214
* TRAP CICS VERSION NUMBER V1.2 (V5.2) @15A 00638209
IF MSGID='DFHSI1500' 00638210
& TOKEN(12) = 'Version' & TOKEN(13)=VERNUM 00638211
THEN EXEC(CMD('EVESROUT EVEEI004 'VERNUM) ROUTE(ALL *)) 0
DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638213
* 00638214
* CICS STARTUP TYPE (COLD/WARM/EMER,ETC.) 00638215
IF (MSGID='DFH1500' | MSGID='DFH1502') & 00638216
TOKEN(3)='CICS' & TOKEN(4) = 'START-UP' & 00638217
TOKEN(5)='IS' & TOKEN(6) = STARTTYP 00638218
THEN EXEC(CMD('EVESROUT EVEEI006 'STARTTYP) ROUTE(ALL *)) 00638219
DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638220
* 00638221
* CICS STARTUP TYPE (COLD/WARM/EMER,ETC.) 00638222
IF MSGID='DFHSI1502I' & 00638223
TOKEN(6) = STARTTYP 00638224
THEN EXEC(CMD('EVESROUT EVEEI006 'STARTTYP) ROUTE(ALL *)) 00638225
DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638226
* 00638227
*
* CICS FIRST OCCURRENCE OF APPLID @12C 4@14D
* HANDLE DFH1280 WITH AND WITHOUT DASH. WHEN THE CICS IS 11@14A
* DEFINED AS A FORMAL MVS SUBSYSTEM IN SYS1.PARMLIB(IEFSSNxx) WITH A
* SUBSYSTEM INITIALIZATION ROUTINE OF DFHSSIN, THE DASH (-) IS REMOVED.
IF MSGID='DFH1280' & TOKEN(2) = '-'
& TOKEN(3) = APPLID
THEN EXEC(CMD('EVESROUT EVEEI007 'APPLID) ROUTE(ALL *))
DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
IF MSGID='DFH1280' & TOKEN(2) ^= '-'
& TOKEN(2) = APPLID

```



|                                                                |                  |
|----------------------------------------------------------------|------------------|
| THEN EXEC(CMD('EVESROUT EVEEI007 'APPLID) ROUTE(ALL *))        |                  |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                |                  |
| *                                                              | 00638233         |
| IF MSGID='DFHTD0100I'                                          | 00638234         |
| & TOKEN(2) = APPLID                                            | 00638235         |
| THEN EXEC(CMD('EVESROUT EVEEI007 'APPLID) ROUTE(ALL *))        | 00638236         |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638237         |
| * CICS STARTUP XRF GENERIC APPLID                              | 00638238         |
| IF (MSGID='DFH6400' .   MSGID = 'DFH6500' .)                   | 00638239         |
| & TEXT= . 'GENERIC APPLID ' USERVAR .                          | 00638240         |
| THEN EXEC(CMD('EVESROUT EVEEI011 'USERVAR) ROUTE(ALL *))       | 00638241         |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638242         |
| * CICS STARTUP XRF GENERIC APPLID                              | 00638243         |
| IF (MSGID='DFHXG6400' .   MSGID = 'DFHXG6500' .)               | 00638244         |
| & TEXT= . 'generic APPLID ' USERVAR .                          | 00638245         |
| THEN EXEC(CMD('EVESROUT EVEEI011 'USERVAR) ROUTE(ALL *))       | 00638246         |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638247         |
| *                                                              | 00638248         |
| * CICS STARTUP TRAP CICS XRFSTATUS                             | 00638249         |
| IF (MSGID='DFH6401' .                                          | 00638250         |
| MSGID='DFH6501' . )                                            | 00638251         |
| & TEXT= . 'CAVM AS ' XRFSTATUS .                               | 00638252         |
| THEN EXEC(CMD('EVESROUT EVEEI005 'XRFSTATUS) ROUTE(ALL *))     | 00638253         |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638254         |
| *                                                              | 00638255         |
| * CICS STARTUP TRAP CICS XRFSTATUS                             | 00638256         |
| IF (MSGID='DFHXG6401' .                                        | 00638257         |
| MSGID='DFHXG6501' . )                                          | 00638258         |
| & TEXT= . 'CAVM as ' XRFSTATUS .                               | 00638259         |
| THEN EXEC(CMD('EVESROUT EVEEI005 'XRFSTATUS) ROUTE(ALL *))     | 00638260         |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638261         |
| *                                                              | 00638262         |
| * CICS TAKEOVER IN PROGRESS                                    | PN34918 00638263 |
| *                                                              | 00638264         |
| IF MSGID='DFH6513'. & TEXT = . 'TAKEOVER REQUEST ACCEPTED' .   | 00638265         |
| THEN EXEC(CMD('EVESROUT EVEEI005 TAKEOVER') ROUTE(ALL *))      | 00638266         |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638267         |
| *                                                              | 3011A 00638264   |
| IF MSGID='DFHXG6513'. & TEXT = . 'TAKEOVER REQUEST ACCEPTED' . | 00638265         |
| THEN EXEC(CMD('EVESROUT EVEEI005 TAKEOVER') ROUTE(ALL *))      | 00638266         |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638267         |
| *                                                              | 00638268         |
| IF MSGID='DFHXG6513'. & TEXT = . 'Takeover request accepted' . | 00638269         |
| THEN EXEC(CMD('EVESROUT EVEEI005 TAKEOVER') ROUTE(ALL *))      | 00638270         |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638271         |
| *                                                              | 00638272         |
| *                                                              | 00638273         |
| * CICS STARTUP SPECIFY ALTERNATIVE SIT PARMS                   | 00638274         |
| IF MSGID='DFH1500' &                                           | 00638275         |
| TEXT = . 'CONTINUE - ENTER .END' .                             | 00638276         |
| THEN EXEC(CMD('EVESROUT EVEEI115') ROUTE(ALL *))               | 00638277         |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638278         |
| *                                                              | 00638279         |
| * CICS STARTUP SPECIFY ALTERNATIVE SIT PARMS                   | @11c 00638280    |
| IF MSGID='DFHPA1104'                                           | 00638281         |
| MSGID='DFHPA1105'                                              | 00638282         |
| MSGID='DFHPA1912'                                              | 00638283         |
| MSGID='DFHPA1915'                                              | 00638284         |
| MSGID='DFHPA1918'                                              | 00638285         |
| THEN EXEC(CMD('EVEEAOPS') ROUTE(ALL *))                        | 00638286         |
| EXEC(CMD('EVESROUT EVEEI115') ROUTE(ALL *))                    | 00638286         |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638287         |
| *                                                              | 00638288         |
| * CICS STARTUP PLTP1 NOT FOUND                                 | 00638289         |

|                                                                |          |
|----------------------------------------------------------------|----------|
| IF MSGID='DFH1578'                                             | 00638290 |
| THEN EXEC(CMD('EVESROUT EHKEGRP AUTOTYP=START')) ROUTE(ALL *)  | 00638291 |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638292 |
| *                                                              | 00638293 |
| * CICS STARTUP PLTPI NOT FOUND                                 | 00638294 |
| * added period '.' after message for trailing D @06A           | 00638295 |
| IF MSGID='DFHSI1578'.                                          | 00638296 |
| THEN EXEC(CMD('EVESROUT EHKEGRP AUTOTYP=START')) ROUTE(ALL *)  | 00638297 |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638298 |
| *                                                              | 00638299 |
| * CICS STARTUP PLTPI PROGRAM NOT FOUND                         | 00638300 |
| IF MSGID='DFH1579'                                             | 00638301 |
| THEN EXEC(CMD('EVESROUT EHKEGRP AUTOTYP=START')) ROUTE(ALL *)  | 00638302 |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638303 |
| *                                                              | 00638304 |
| * CICS STARTUP PLTPI PROGRAM NOT FOUND                         | 00638305 |
| * added period '.' after message for trailing D @06A           | 00638306 |
| IF MSGID='DFHSI1579'.                                          | 00638307 |
| THEN EXEC(CMD('EVESROUT EHKEGRP AUTOTYP=START')) ROUTE(ALL *)  | 00638308 |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638309 |
| *                                                              | 00638304 |
| * CICS ABENDED PLTPI PROGRAM @15A                              | 00638305 |
| * reply CANCEL to wtor                                         | 00638306 |
| IF MSGID='DFHSI1580'.                                          | 00638307 |
| THEN EXEC(CMD('EVESROUT EHKEGRP AUTOTYP=START')) ROUTE(ALL *)  | 00638308 |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638309 |
| *                                                              | @16A     |
| * CICS STARTUP - CICS TS ARM REGISTER FAILED... A REPLY IS     | @16A     |
| * REQUIRED                                                     | @16A     |
| IF MSGID='DFHKE0408D' & TEXT = MESSAGE                         |          |
| THEN EXEC(CMD('EVESROUT EVEEARMW 'MESSAGE) ROUTE(ALL *))       |          |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                |          |
| *                                                              | @16A     |
| * CICS RLS OFFSITE=YES STARTUP - RECOVERY CO-ORDINATION        | @16A     |
| * NOTE: UNCOMMENT FOLLOWING 3 STMTS TO USE CICS TS PROVIDED    | @16A     |
| * REXX TO CO-ORDINATE VSAM RLS OFFSITE RECOVERY.               | @16A     |
| *IF (MSGID='DFHFC0574'.   MSGID='DFHFC0575'.) & TEXT = MESSAGE |          |
| * THEN EXEC(CMD('EVESROUT DFH\$OFAR 'MESSAGE) ROUTE(ALL *))    |          |
| * DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);              |          |
| *                                                              | 00638310 |
| * CICS STARTUP NUCLEUS MODULE NOT FOUND - NEEDS REPLY @03      | 00638311 |
| IF MSGID='DFH1596' & TEXT = MESSAGE                            | 00638312 |
| & HDRMTYPE = '>'                                               | 00638313 |
| THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))       | 00638314 |
| EXEC(CMD('EVESROUT EHKEGRP AUTOTYP=START')) ROUTE(ALL *)       | 00638315 |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638316 |
| *                                                              | 00638317 |
| * CICS STARTUP NUCLEUS MODULE NOT FOUND - INFORMATIONAL @03    | 00638318 |
| IF MSGID='DFH1596' & TEXT = MESSAGE                            | 00638319 |
| & HDRMTYPE = '>'                                               | 00638320 |
| THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))       | 00638321 |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638322 |
| *                                                              | 00638323 |
| * CICS STARTUP NUCLEUS MODULE NOT FOUND - NEEDS REPLY @03      | 00638324 |
| IF MSGID='DFHSI1596' & TEXT = MESSAGE                          | 00638325 |
| & HDRMTYPE = '>'                                               | 00638326 |
| THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))       | 00638327 |
| EXEC(CMD('EVESROUT EHKEGRP AUTOTYP=START')) ROUTE(ALL *)       | 00638328 |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638329 |
| *                                                              | 00638330 |
| * CICS STARTUP XRF STATUS STANDBY                              | 00638331 |
| IF (MSGID='DFH1041' .                                          | 00638332 |
| & JOBNAME = SVJOB )                                            | 00638333 |
| THEN EXEC(CMD('EVESROUT EVEEI008 'SVJOB) ROUTE(ALL *))         | 00638334 |

|                                                                |                 |
|----------------------------------------------------------------|-----------------|
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638335        |
| *                                                              | 00638336        |
| * CICS STARTUP XRF STATUS STANDBY                              | 00638337        |
| IF (MSGID='DFHTC1041' .                                        | 00638338        |
| & JOBNAME = SVJOB )                                            | 00638339        |
| THEN EXEC(CMD('EVESROUT EVEEI008 'SVJOB) ROUTE(ALL *)))        | 00638340        |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638341        |
| *                                                              | 00638342        |
| * CICS NOW ACTIVE                                              | 00638343        |
| * VTAMACB STATE ACTION - CICS UP                               | 00638344        |
| IF (MSGID='DFH1500'                                            | 00638345        |
| MSGID='DFH1517')                                               | 00638346        |
| & TEXT= .'CONTROL IS BEING GIVEN TO CICS' .                    | 00638347        |
| THEN EXEC(CMD('EVESROUT EVEEI009 ') ROUTE(ALL *)))             | 00638348        |
| EXEC(CMD('EVESROUT EVEEY00S MSGID=CICSUP') ROUTE(ALL *)))      | 00638349        |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638350        |
| *                                                              | 00638351        |
| * CICS NOW ACTIVE                                              | 00638352        |
| * VTAMACB STATE ACTION - CICS UP                               | 00638353        |
| IF MSGID='DFHSI1517'                                           | 00638354        |
| THEN EXEC(CMD('EVESROUT EVEEI009 ') ROUTE(ALL *)))             | 00638355        |
| EXEC(CMD('EVESROUT EVEEY00S MSGID=CICSUP') ROUTE(ALL *)))      | 00638356        |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638357        |
| *                                                              | 00638358        |
| * -----*                                                       | 00638359        |
| * CICS SHUTDOWN MESSAGE AUTOMATION *                           | 00638360        |
| * -----*                                                       | 00638361        |
| *                                                              | 00638362        |
| * SHUTDOWN HAS BEEN INITIATED                                  | 00638363        |
| IF (MSGID='DFH1701'                                            | 00638364        |
| MSGID='DFH1702'                                                | 00638365        |
| MSGID='DFH1703'                                                | 00638366        |
| MSGID='DFH1704'                                                | 00638367        |
| MSGID='DFH1713' )                                              | 00638368        |
| THEN EXEC(CMD('EVESROUT EVEET001 ') ROUTE(ALL *)))             | 00638369        |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638370        |
| *                                                              | 00638371        |
| * SHUTDOWN HAS BEEN INITIATED                                  | 00638372        |
| IF (MSGID='DFHTM1701'                                          | 00638373        |
| MSGID='DFHTM1702'                                              | 00638374        |
| MSGID='DFHTM1713' )                                            | 00638375        |
| THEN EXEC(CMD('EVESROUT EVEET001 ') ROUTE(ALL *)))             | 00638376        |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638377        |
| *                                                              | 00638378        |
| * CICS SHUTDOWN - UPDATE WARM KEYPOINT STATUS                  | @15C * 00638379 |
| IF (MSGID='DFH1796I'                                           | 00638380        |
| MSGID='DFHRM0130'                                              | 00638380        |
| MSGID='DFHWK0105I')                                            | 00638381        |
| THEN EXEC(CMD('EVESROUT EVEET002 ') ROUTE(ALL *)))             | 00638382        |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638383        |
| *                                                              | 00638380        |
| * CICS SHUTDOWN - IN DOUBT UNITS OF WORK UPDATE                | @15A * 00638379 |
| IF (MSGID='DFHRM0203'                                          | 00638380        |
| MSGID='DFHRM0204')                                             | 00638381        |
| THEN EXEC(CMD('EVESROUT EVEETUOW ') ROUTE(ALL *)))             | 00638382        |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                | 00638383        |
| *                                                              | 00638380        |
| * CICS SHUTDOWN - LOG ERROR MANUAL SHUTDOWN REQUESTED          | @15A * 00638379 |
| IF (MSGID='DFHLG0739'                                          | 00638380        |
| MSGID='DFHLG0750') & JOBNAME = SVJOB & TEXT = MESSAGE          | 0063838         |
| THEN EXEC(CMD('EVESROUT CICSSHUT NORMAL 'SVJOB) ROUTE(ALL *))) |                 |
| EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *)))           | 00638654        |

```

 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638383
* 00638380
* CICS - CRITICAL MSG for SDF @15A * 00638379
IF (MSGID='DFHLG0507' | 00638380
 MSGID='DFHLG0730' | 00638380
 MSGID='DFHLG0731' | 00638380
 MSGID='DFHLG0734' | 00638380
 MSGID='DFHLG0735') & TEXT = MESSAGE 00638380
 THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *)) 00638653
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638655
* 00638380
* CICS - CRITICAL MSG for SDF/INITIAL START REQUIRED @15A * 00638379
IF (MSGID='DFHLG0736' | 00638380
 MSGID='DFHLG0738' | 00638380
 MSGID='DFHLG0740') & TEXT = MESSAGE 00638380
 THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *)) 00638653
 EXEC(CMD('EVESROUT EVEETUOW ') ROUTE(ALL *)) 00638654
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638655
* 00638380
* CICS SHUTDOWN - INITIAL START REQUIRED UPDATE @15A * 00638379
IF (MSGID='DFHRM0144' | 00638380
 MSGID='DFHDM0106' | 00638380
 MSGID='DFHSI1542') & TOKEN(2) = CICSNAME 00638381
 THEN EXEC(CMD('EVESROUT EVEETUOW ') ROUTE(ALL *)) 00638382
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638383
* 00638384
* CICS SHUTDOWN - XRF SIGNOFF 00638385
IF (MSGID='DFH6405' . | 00638386
 MSGID='DFH6409' .) 00638387
 THEN EXEC(CMD('EVESROUT EVEET005 ') ROUTE(ALL *)) 00638388
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638389
* 00638390
* CICS SHUTDOWN - XRF SIGNOFF 00638391
IF (MSGID='DFHXG6405' . | 00638392
 MSGID='DFHXG6409' .) 00638393
 THEN EXEC(CMD('EVESROUT EVEET005 ') ROUTE(ALL *)) 00638394
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638395
* -----* 00638396
* CICS TRANSACTION ABENDS * 00638397
* -----* 00638398
* 00638399
* TRANSACTION RECOVERY @16A 00638409
*
IF (MSGID='DFHAC2231' . | MSGID='DFHAC2232' . | MSGID='DFHAC2233' . | 00638410
 MSGID='DFHAC2236' . | MSGID='DFHAC2237' . | MSGID='DFHAC2238' . | 00638410
 MSGID='DFHAC2245' . | MSGID='DFHAC2246' . | MSGID='DFHAC2247' . | 00638410
 MSGID='DFHAC2248' . | MSGID='DFHAC2249' . | MSGID='DFHAC2250' . | 00638410
 MSGID='DFHAC2251' . | MSGID='DFHAC2252' . | MSGID='DFHAC2253' .) 00638410
 THEN EXEC(CMD('EVESROUT EVEERTRN') 00638414
 ROUTE(ALL *)) 00638415
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638416
* 00638399
* TRANSACTION RECOVERY 00638409
*
IF ((MSGID='DFH2236' . | MSGID='DFH2237' . | MSGID='DFH2238' .) & 00638401
 TOKEN(3)=TRANID & 00638402
 TOKEN(5)=ABCODE & 00638403
 TOKEN(8)=PGMID .) 00638404
 THEN EXEC(CMD('EVESROUT EVEERTRN 'TRANID' 'ABCODE' 'PGMID') 00638405
 ROUTE(ALL *)) 00638406
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638407
* -----* 00638417
* CICS SYSTEM ABENDS * 00638418
* -----* 00638419

```

|                                                         |                 |
|---------------------------------------------------------|-----------------|
| *                                                       | 00638420        |
| * CICS REGION ABEND RECOVERY                            | 00638421        |
| IF MSGID='DFH0401' &                                    | 00638422        |
| TEXT = . 'ABEND' ABCODE .                               | 00638423        |
| THEN EXEC(CMD('EVESROUT EVEET003 'ABCODE) ROUTE(ALL *)) | 00638424        |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);         | 00638425        |
| *                                                       | 00638426        |
| * CICS REGION ABEND RECOVERY                            | 00638427        |
| IF MSGID='DFHPC0401' &                                  | 00638428        |
| TEXT = . 'Abend' ABCODE .                               | 00638429        |
| THEN EXEC(CMD('EVESROUT EVEET003 'ABCODE) ROUTE(ALL *)) | 00638430        |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);         | 00638431        |
| *                                                       | 00638432        |
| * CICS REGION ABEND RECOVERY                            | 00638433        |
| IF MSGID='DFH0405' &                                    | 00638434        |
| TEXT = . 'PROCESSING ABEND' ABCODE .                    | 00638435        |
| THEN EXEC(CMD('EVESROUT EVEET003 'ABCODE) ROUTE(ALL *)) | 00638436        |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);         | 00638437        |
| *                                                       | 00638438        |
| * CICS REGION ABEND RECOVERY                            | 00638439        |
| IF MSGID='DFHPC0405' &                                  | 00638440        |
| TEXT = . 'processing abend' ABCODE .                    | 00638441        |
| THEN EXEC(CMD('EVESROUT EVEET003 'ABCODE) ROUTE(ALL *)) | 00638442        |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);         | 00638443        |
| *                                                       | 00638444        |
| * CICS REGION ABEND RECOVERY                            | 00638445        |
| IF MSGID='DFH0408' &                                    | 00638446        |
| TEXT = . 'ABEND' ABCODE .                               | 00638447        |
| THEN EXEC(CMD('EVESROUT EVEET003 'ABCODE) ROUTE(ALL *)) | 00638448        |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);         | 00638449        |
| *                                                       | 00638450        |
| * CICS REGION ABEND RECOVERY                            | 00638451        |
| IF MSGID='DFHPC0408' &                                  | 00638452        |
| TEXT = . 'Abend' ABCODE .                               | 00638453        |
| THEN EXEC(CMD('EVESROUT EVEET003 'ABCODE) ROUTE(ALL *)) | 00638454        |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);         | 00638455        |
| *                                                       | 00638456        |
| * CICS REGION ABEND RECOVERY                            | 00638457        |
| IF MSGID='DFH0409' &                                    | 00638458        |
| TEXT = . 'PROCESSING ABEND' ABCODE .                    | 00638459        |
| THEN EXEC(CMD('EVESROUT EVEET003 'ABCODE) ROUTE(ALL *)) | 00638460        |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);         | 00638461        |
| *                                                       | 00638462        |
| * CICS REGION ABEND RECOVERY                            | 00638463        |
| IF MSGID='DFHPC0409' &                                  | 00638464        |
| TEXT = . 'processing abend' ABCODE .                    | 00638465        |
| THEN EXEC(CMD('EVESROUT EVEET003 'ABCODE) ROUTE(ALL *)) | 00638466        |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);         | 00638467        |
| *                                                       | 00638468        |
| * CICS REGION ABEND -- DFH0409 ISSUED AS MLWTO          | @09A*/ 00638469 |
| IF MSGID='DFH0409'   MSGID='DFHPC0409'                  | 00638470        |
| THEN EXEC(CMD('EVESROUT EVEET003 ') ROUTE(ALL *))       | 00638471        |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);         | 00638472        |
| *                                                       | 00638473        |
| * CICS REGION ABEND RECOVERY                            | 00638474        |
| IF MSGID='DFH0501'                                      | 00638475        |
| MSGID='DFH0504'                                         | 00638476        |
| MSGID='DFH0601'                                         | 00638477        |
| MSGID='DFHSR0601'                                       | 00638478        |
| MSGID='DFH0602'                                         | 00638479        |
| MSGID='DFHSR0602'                                       | 00638480        |
| MSGID='DFH0603'                                         | 00638481        |
| MSGID='DFHSR0603'                                       | 00638482        |
| MSGID='DFH0607'                                         | 00638483        |

|                                                                   |        |          |
|-------------------------------------------------------------------|--------|----------|
| MSGID='DFH0612'                                                   |        | 00638484 |
| MSGID='DFHSR0612'                                                 |        | 00638485 |
| THEN EXEC(CMD('EVESROUT EVEET003')) ROUTE(ALL *)                  |        | 00638486 |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                   |        | 00638487 |
| *                                                                 |        | 00638488 |
| * CICS REGION ABEND RECOVERY (CONTINUED)                          |        | 00638489 |
| IF MSGID='DFH0613'                                                |        | 00638490 |
| MSGID='DFHSR0613'                                                 |        | 00638491 |
| MSGID='DFH0614'                                                   |        | 00638492 |
| MSGID='DFH0615'                                                   |        | 00638493 |
| MSGID='DFHSR0615'                                                 |        | 00638494 |
| MSGID='DFH0619'                                                   |        | 00638495 |
| MSGID='DFH0620'                                                   |        | 00638496 |
| THEN EXEC(CMD('EVESROUT EVEET003')) ROUTE(ALL *)                  |        | 00638497 |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                   |        | 00638498 |
| *                                                                 |        | 00638499 |
| * CICS GENERIC ABEND                                              |        | 00638500 |
| IF MSGID='DFH0606' &                                              |        | 00638501 |
| TEXT = . 'ABEND' ABCODE .                                         |        | 00638502 |
| THEN EXEC(CMD('EVESROUT EVEET003 'ABCODE) ROUTE(ALL *))           |        | 00638503 |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                   |        | 00638504 |
| *                                                                 |        | 00638505 |
| * CICS GENERIC ABEND VERSION 3.3                                  |        | 00638506 |
| IF MSGID='DFHSR0606' &                                            |        | 00638507 |
| TEXT = . 'Abend' . '/' ABCODE ' )' .                              |        | 00638508 |
| THEN EXEC(CMD('EVESROUT EVEET003 'ABCODE) ROUTE(ALL *))           |        | 00638509 |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                   |        | 00638510 |
| *                                                                 |        | 00638511 |
| * CICS REGION ABEND RECOVERY                                      | @15C * | 00638512 |
| IF MSGID='DFH1797I'                                               |        | 00638513 |
| MSGID='DFHTM1797I'                                                |        | 00638514 |
| MSGID='DFHTM1797'                                                 |        | 00638514 |
| MSGID='DFH3784'                                                   |        | 00638515 |
| THEN EXEC(CMD('EVESROUT EVEET003')) ROUTE(ALL *)                  |        | 00638516 |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                   |        | 00638517 |
| *                                                                 |        | 00638518 |
| * CICS REGION ABEND RECOVERY                                      |        | 00638519 |
| IF MSGID='DFHSR0605'                                              |        | 00638520 |
| MSGID='DFH1306'                                                   |        | 00638521 |
| MSGID='DFHTS1306'                                                 |        | 00638522 |
| MSGID='DFH5725'                                                   |        | 00638523 |
| MSGID='DFHER5725'                                                 |        | 00638524 |
| THEN EXEC(CMD('EVESROUT EVEET003')) ROUTE(ALL *)                  |        | 00638525 |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                   |        | 00638526 |
| *                                                                 |        | 00638527 |
| * CICS REGION ABEND RECOVERY                                      |        | 00638528 |
| IF MSGID='DFH0614'                                                |        | 00638529 |
| MSGID='DFH1001' .                                                 |        | 00638530 |
| MSGID='DFHTC1001' .                                               |        | 00638531 |
| THEN EXEC(CMD('EVESROUT EVEET003')) ROUTE(ALL *)                  |        | 00638532 |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                   |        | 00638533 |
| *                                                                 |        | 00638534 |
| * CICS REGION ABEND RECOVERY                                      |        | 00638535 |
| IF MSGID='DFHCC0001' . &                                          |        | 00638536 |
| TEXT = . '(code' ABCODE '/' ABCODE1 ' )' .                        |        | 00638537 |
| THEN EXEC(CMD('EVESROUT EVEET003 'ABCODE' 'ABCODE1) ROUTE(ALL *)) |        | 00638538 |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                   |        | 00638539 |
| *                                                                 |        | 00638540 |
| * CICS REGION ABEND RECOVERY                                      |        | 00638541 |
| IF MSGID='DFHKE1800'                                              |        | 00638542 |
| THEN EXEC(CMD('EVESROUT EVEET003')) ROUTE(ALL *)                  |        | 00638543 |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);                   |        | 00638544 |
| *                                                                 |        | 00638545 |
| * -----*                                                          |        | 00638546 |

```

* CICS STATE/ACTION MESSAGE ENTRIES * 00638547
----- 00638548
* 00638549
* AUTOINSTALL STATE ACTION 00638550
* removed I after message DFHZC5966 @07A 00638551
IF (MSGID='DFH3482' | MSGID='DFHZC3482' |
 MSGID='DFH3483' | MSGID='DFHZC3483' |
 MSGID='DFH5966I' | MSGID='DFHZC5966' |
 MSGID='DFH5935I') 00638552
 00638553
 00638554
 00638555
THEN EXEC(CMD('EVESROUT EVEEY00S ') ROUTE(ALL *)) 00638556
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638557
* 00638558
* STORAGE VIOLATIONS STATE ACTION 00638559
IF (MSGID='DFH0508' | MSGID='DFHSM0102') 00638560
THEN EXEC(CMD('EVESROUT EVEEY00S ') ROUTE(ALL *)) 00638561
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638562
* 00638563
* SOS STATE ACTION PN36057 PN52486 00638564
IF (MSGID='DFH0506' . | MSGID='DFHSM0100' | 00638565
 MSGID='DFH0507' . | MSGID='DFHSM0101' | 00638566
 MSGID='DFHSM0106' | MSGID='DFHSM0107' | 00638567
 MSGID='DFHSM0116' | MSGID='DFHSM0117' | 00638568
 MSGID='DFHSM0131' . | MSGID='DFHSM0132' . | 00638569
 MSGID='DFHSM0133' . | MSGID='DFHSM0134' .) 00638570
THEN EXEC(CMD('EVESROUT EVEEY00S ') ROUTE(ALL *)) 00638571
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638572
* 00638573
* VTAMACB STATE ACTION - ACB CLOSED BY CICS 00638574
IF MSGID = 'DFH2316I' 00638575
 THEN EXEC(CMD('EVESROUT EVEEY00S MSGID=ACBCLS') ROUTE(ALL *)); 00638576
* 00638577
* VTAMACB STATE ACTION - ACB CLOSED BY CICS 00638578
IF MSGID = 'DFHZC2316' 00638579
 THEN EXEC(CMD('EVESROUT EVEEY00S MSGID=ACBCLS') ROUTE(ALL *)); 00638580
* 00638581
* STATE ACTION WHEN VTAMACB OPENED 00638582
IF MSGID = 'DFHZC3463' 00638583
 THEN EXEC(CMD('EVESROUT EVEES414') ROUTE(ALL *)); 00638584
* 00638585
* VTAMACB STATE ACTION - VTAM DISCOVERED DOWN BY CICS 00638586
IF (MSGID = 'DFH3463I' 00638587
 | MSGID = 'DFHSI1572' 00638588
 | MSGID = 'DFH1572') 00638589
 & (TEXT = . 'X''52''' . | TEXT= . 'X''5C''' . 00638590
 | TEXT = . '=52.' . | TEXT= . '=5C.' .) 00638591
 THEN EXEC(CMD('EVESROUT EVEEY00S MSGID=VTAMDN') ROUTE(ALL *)); 00638592
* 00638593
* DFH1572 removed as DFH1592 will do the job PN32138 00638594
* 00638595
* VTAMACB STATE ACTION - VTAM MAJOR NODE ERROR FROM CICS 00638596
* DFH1572 REMOVED AS DFH1592 WILL DO THE JOB 00638597
* @08A 00638598
IF MSGID = 'DFH3463I' 00638599
 & (TEXT = . 'X''5A''' . 00638600
 | TEXT = . '=5A.' .) 00638601
 THEN EXEC(CMD('EVESROUT EVEEY00S MSGID=MAJNER') ROUTE(ALL *)); 00638602
* 00638603
* VTAMACB STATE ACTION - VTAM MAJOR NODE ERROR FROM CICS 00638604
* REQUIRED FOR XRF ALTERNATE IN ANOTHER CEC PN32138 00638605
IF (MSGID = 'DFH1592I' 00638606
 | MSGID = 'DFHSI1592') 00638607
 THEN EXEC(CMD('EVESROUT EVEEY00S MSGID=MAJNER') ROUTE(ALL *)); 00638608
* 00638609
* VTAMACB STATE ACTION - ACB WAS OPENED BY CICS 00638610

```

```

IF MSGID = 'DFH3463I' 00638611
 & (TEXT = . 'X''00'' . 00638612
 | TEXT = . '=00.' .) 00638613
 THEN EXEC(CMD('EVESROUT EVEEY00S MSGID=ACBOPN') ROUTE(ALL *)); 00638614
* 00638615
* DFH1572 with RC = 5A is the same error as DFH1592 PN32138 00638616
* 00638617
* VTAMACB STATE ACTION - ALL OTHER CASES, VTAM ERROR 00638618
IF (MSGID = 'DFH3463I' 00638619
 | MSGID = 'DFHSI1572' 00638620
 | MSGID = 'DFH1572') 00638621
 & (TEXT ^= . 'X''5A'' . 00638622
 & TEXT ^= . '=5A.' .) 00638623
 THEN EXEC(CMD('EVESROUT EVEEY00S MSGID=VTAMER') ROUTE(ALL *)); 00638624
* 00638625
* -----* 00638626
* CICS CRITICAL MESSAGE PROCESSING * 00638627
* -----* 00638628
* 00638629
IF MSGID='DFH0964' & TEXT = MESSAGE 00638630
 THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) 00638631
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638632
* 00638633
IF MSGID='DFH0978' & TEXT = MESSAGE 00638634
 THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) 00638635
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638636
* 00638637
IF MSGID='DFH0987' & TEXT = MESSAGE 00638638
 THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) 00638639
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638640
* 00638641
* 1307 V2-RPY CANCEL? 00638642
IF MSGID='DFH1307' & TEXT = MESSAGE 00638643
 THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) 00638644
 EXEC(CMD('EVESROUT EHKEGRP AUTOTYP=RECOVERY') ROUTE(ALL *))) 00638645
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638646
* 1307 V3-RPY CANCEL? - MSG retired for CICS V5+ @15C 00638647
* If running CICS V5+ this message should be removed
IF (MSGID='DFHTS1307') & TEXT = MESSAGE 00638648
 THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) 00638649
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638650
* 1308 V2-Information - MSG retired for CICS V5+ @10c @15C 00638651
* If running CICS V5+ this message should be removed
IF MSGID='DFH1308' & TEXT = MESSAGE 00638652
 & HDRMTYPE ^= '>'
 THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *)))
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
* 1308 V2,V3-RPY CANCEL? - MSG retired for CICS V5+ @10a @15C
* If running CICS V5+ this message should be removed
IF MSGID='DFH1308' & TEXT = MESSAGE
 & HDRMTYPE = '>'
 THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) 00638653
 EXEC(CMD('EVESROUT EHKEGRP AUTOTYP=RECOVERY') ROUTE(ALL *))) 00638654
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638655
* 1308 V4-RPY CANCEL? - MSG retired for CICS V5+ @10c @15C 00638656
* If running CICS V5+ this message should be removed
IF MSGID='DFHTS1308' & TEXT = MESSAGE 00638657
 & HDRMTYPE = '>'
 THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) 00638658
 EXEC(CMD('EVESROUT EHKEGRP AUTOTYP=RECOVERY') ROUTE(ALL *))) 00638659
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00638660
* 00638661
IF MSGID='DFH1310' & TEXT = MESSAGE 00638662
 THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) 00638663

```



|                                                           |               |
|-----------------------------------------------------------|---------------|
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           | 00638664      |
| *                                                         | 00638665      |
| IF MSGID='DFHTS1310' & TEXT = MESSAGE                     | 00638666      |
| THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) | 00638667      |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           | 00638668      |
| *                                                         | 00638669      |
| IF MSGID='DFH1311' & TEXT = MESSAGE                       | 00638670      |
| THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) | 00638671      |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           | 00638672      |
| *                                                         | 00638673      |
| IF MSGID='DFHTS1311' & TEXT = MESSAGE                     | 00638674      |
| THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) | 00638675      |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           | 00638676      |
| *                                                         | 00638677      |
| IF MSGID='DFH1313I' & TEXT = MESSAGE                      | 00638678      |
| THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) | 00638679      |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           | 00638680      |
| - MSG retired for CICS V5+                                | @15C 00638681 |
| * If running CICS V5+ this message should be removed      |               |
| IF MSGID='DFHTS1313' & TEXT = MESSAGE                     | 00638682      |
| THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) | 00638683      |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           | 00638684      |
| *                                                         | 00638685      |
| IF MSGID='DFH2104' & TEXT = MESSAGE                       | 00638686      |
| THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) | 00638687      |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           | 00638688      |
| *                                                         | 00638689      |
| IF MSGID='DFH2108' & TEXT = MESSAGE                       | 00638690      |
| THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) | 00638691      |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           | 00638692      |
| *                                                         | 00638693      |
| IF MSGID='DFH2110' & TEXT = MESSAGE                       | 00638694      |
| THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) | 00638695      |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           | 00638696      |
| *                                                         | 00638697      |
| IF MSGID='DFH2111' & TEXT = MESSAGE                       | 00638698      |
| THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) | 00638699      |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           | 00638700      |
| *                                                         | 00638701      |
| IF MSGID='DFH2112' & TEXT = MESSAGE                       | 00638702      |
| THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) | 00638703      |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           | 00638704      |
| *                                                         | 00638705      |
| IF MSGID='DFH2801I' & TEXT = MESSAGE                      | 00638706      |
| THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) | 00638707      |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           | 00638708      |
| *                                                         | 00638709      |
| IF MSGID='DFHRU2801' . & TEXT = MESSAGE                   | 00638710      |
| THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) | 00638711      |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           | 00638712      |
| *                                                         | 00638713      |
| IF MSGID='DFH2802I' & TEXT = MESSAGE                      | 00638714      |
| THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) | 00638715      |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           | 00638716      |
| *                                                         | 00638717      |
| IF MSGID='DFHRU2802' . & TEXT = MESSAGE                   | 00638718      |
| THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) | 00638719      |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           | 00638720      |
| *                                                         | 00638721      |
| IF MSGID='DFH2803I' & TEXT = MESSAGE                      | 00638722      |
| THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))) | 00638723      |
| DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);           | 00638724      |
| *                                                         | 00638725      |
| IF MSGID='DFHRU2803' . & TEXT = MESSAGE                   | 00638726      |

```

 THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
IF MSGID='DFH3425I' & TEXT = MESSAGE
 THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
IF MSGID='DFH3784I' & TEXT = MESSAGE
 THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
IF MSGID='DFH3785I' & TEXT = MESSAGE
 THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
IF MSGID='DFHIR3785' . & TEXT = MESSAGE
 THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
IF MSGID='DFH3905I' & TEXT = MESSAGE
 THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
IF MSGID='DFH3906I' & TEXT = MESSAGE
 THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
IF MSGID='DFH3907I' & TEXT = MESSAGE
 THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
IF MSGID='DFH4816I' & TEXT = MESSAGE
 THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
IF MSGID='DFH5909I' & TEXT = MESSAGE
 THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
IF MSGID='DFH5916E' & TEXT = MESSAGE
 THEN EXEC(CMD('EVESROUT EVEED003 'MESSAGE) ROUTE(ALL *))
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
*
* -----*
* Transaction Server VSAM RLS Message Handling *
* -----*
*
* RLS Inactive Messages
IF (MSGID='DFHFC0153' | MSGID='DFHFC0501' |
 MSGID='DFHFC0504' | MSGID='DFHFC0508' |
 MSGID='DFHFC0571' | MSGID='DFHFC0563' |
 MSGID='DFHFC0564' | MSGID='DFHFC0565' |
 MSGID='DFHFC0566' | MSGID='DFHFC0567')
 THEN EXEC(CMD('EVESROUT EVEERLSI RLSINACTIVE') ROUTE(ALL *))
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
* RLS Active Messages
IF (MSGID='DFHFC0562' | MSGID='DFHFC0570' |
 MSGID='DFHFC0577')
 THEN EXEC(CMD('EVESROUT EVEERLSI RLSACTIVE') ROUTE(ALL *))
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*
***** @13A*
* @13A*

```

```

* Trap unknown replies @13A* 00388220
* @13A* 00389210
***** 4@13A* 00390200
 IF IFRAUWF1(6) = '1' & DOMAINID = %AOFDOM%
 THEN EXEC(CMD('OUTREP ')ROUTE(ONE %AOFOPWTORS%));
 00391190
 00392180
 00393170

***** 00722000
* END OF CICS AUTOMATION EVEMDFH1 MESSAGE TABLE ENTRIES * 00723000
***** 00724000
 ALWAYS;
 00724100
 END;
 00724200

```

## EVEMCON1—Messages in Conflict with AOC/MVS (Included by CICSMSG1)

```
*****00001000
* (C) COPYRIGHT IBM CORP. 1995 - ALL RIGHTS RESERVED
*
* LICENSED MATERIALS - PROPERTY OF IBM
* RESTRICTED MATERIALS OF IBM
* 5685-151
*
* US GOVERNMENT USERS RESTRICTED RIGHTS -
* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY
* GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
*
*****00004000
*****00005000
* APAR# DATE *
* -----
* $01=OW08596 12/13/94 JS AOC CICS MESSAGE TABLE TRAP FOR MSGDFS3463I*
* INCORRECT -- TRAPS WRONG MESSAGE *
* ADD DISTAUTO TRAP TO IEE115I *
* 1.3 04/25/94 JG CREATED FOR AOC 1.3 *
*
*****00054000
*****00055000
* START OF CICS AUTOMATION OPTION MESSAGE TABLE ENTRIES WITH *
* POSSIBLE CONFLICTS *
*
* ENTRIES WITH "POSSIBLE CONFLICT" AS A COMMENT ARE MESSAGES WITHOUT*
* THE DFH.. OR EVE.. PREFIX AND THAT MAY HAVE ALREADY BEEN SPECIFIED*
* IN THE MESSAGE TABLE. MESSAGES SUCH AS IEF403I AND IEF404I ARE *
* EXAMPLES OF SUCH MESSAGES. CAREFULLY REVIEW THESE MESSAGES FOR *
* CONFLICTING SPECIFICATIONS. *
*
* ENTRIES WITH "SPECIFIC AUTOMATED OPERATOR NAME" AS A COMMENT ARE *
* MVS MESSAGES THAT ARE CRITICAL TO THE PROPER MESSAGE PROCESSING *
* FLOW, BUT WOULD TYPICALLY BE ASSIGNED TO ANOTHER AUTOMATED *
* OPERATOR. MESSAGES SUCH AS IEF403I AND IEF404I ARE EXAMPLES. *
* THESE MESSAGES SHOULD BE ROUTED TO THE AUTOMATED OPERATOR WHICH *
* HANDLES THE DFH* AND EVE* MESSAGES. REVIEW THESE MESSAGE ENTRIES *
* CAREFULLY. *
*****00071000
* CICS JOB STARTED ... TOKEN 2 SHOULD BE THE CICS JOBNAMES 00073000
* CICS AUTOMATION OF VTAMACB - CICS STARTUP IEF403I 00074000
* WARNING ----POSSIBLE CONFLICT WITH EXISTING MESSAGE TABLE STMTS 00075000
* WARNING ----SPECIFIC AUTOMATED OPERATOR NAME ON ROUTE FUNCTION 00076000
IF MSGID='IEF403I' & TOKEN(2)=SVJOB &
(TOKEN(2)='CICS'. |
TOKEN(2)='CICS'.)
THEN EXEC(CMD('EVESROUT JOBNAME='SVJOB' EVEEI002 'SVJOB')
ROUTE(ONE AUTCICS))
DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*****00257000
* CICS SHUTDOWN - JOB ENDED 00258000
* WARNING ----POSSIBLE CONFLICT WITH EXISTING MESSAGE TABLE STMTS 00259000
* WARNING ----SPECIFIC AUTOMATED OPERATOR NAME ON ROUTE FUNCTION 00260000
IF MSGID='IEF404I' & TOKEN(2) = SVJOB &
(TOKEN(2)='CICS'. |
TOKEN(2)='CICS'.)
THEN EXEC(CMD('EVESROUT JOBNAME='SVJOB' EVEET004 'SVJOB')
ROUTE(ONE AUTCICS))
DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);
*****00268000
```

```

* CICS SHUTDOWN - JOB ENDED 00269000
* WARNING ----POSSIBLE CONFLICT WITH EXISTING MESSAGE TABLE STMTS 00270000
* WARNING ----SPECIFIC AUTOMATED OPERATOR NAME ON ROUTE FUNCTION 00271000
IF MSGID='IEF450I' & TOKEN(2) = SVJOB & 00272000
 TEXT = . 'ABEND=' ABCODE ABCODE1 . & 00273000
 (TOKEN(2)='CICS'. | 00274000
 TOKEN(2)='CICS'.) 00276000
 THEN EXEC(CMD('EVESROUT JOBNAME='SVJOB' EVEET003 'ABCODE' 'ABCODE1) 00277000
 ROUTE(ONE AUTCICS)) 00278000
 EXEC(CMD('EVESROUT JOBNAME='SVJOB' EVEET004 'SVJOB) 00279000
 ROUTE(ONE AUTCICS)) 00280000
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00281000
* 00480000
* AUTOMATION OF VTAMACB - VTAM UP 00481000
* WARNING ----POSSIBLE CONFLICT WITH EXISTING MESSAGE TABLE STMTS 00482000
IF MSGID='IST020I' 00483000
 & TEXT=MESSAGE 00484000
 THEN EXEC(CMD('EVEES408') ROUTE(ALL *))) 00485000
 CONTINUE(Y); 00281000
* 00486000
* CICS AUTOMATION OF VTAMACB - VTAM DOWN 00487000
* WARNING ----POSSIBLE CONFLICT WITH EXISTING MESSAGE TABLE STMTS 00488000
IF MSGID='IST102I' 00489000
 & TEXT=MESSAGE 00490000
 THEN EXEC(CMD('EVEES409') ROUTE(ALL *))) 00491000
 CONTINUE(Y); 00281000
* 00492000
* 00510000
* CICS AUTOMATION OF VTAM CDRM RECOVERY 00511000
* WARNING ----POSSIBLE CONFLICT WITH EXISTING MESSAGE TABLE STMTS 00512000
IF MSGID='IST093I' & TOKEN(2)=RESOURCE . 00513000
 THEN EXEC(CMD('EVEEZ000 ' RESOURCE) ROUTE(ALL *)); 00514000
* 00515000
* 00657000
* -----* 00658000
* CICS STARTUP CONTROL * 00659000
* -----* 00660000
* 00661000
* MVS DISPLAY ACTIVE 00662000
* WARNING ----POSSIBLE CONFLICT WITH EXISTING MESSAGE TABLE STMTS 00663000
* ADDED DISTAUTO @01A
IF (MSGID='IEE105I' | MSGID='IEE115I') & TEXT=MESSAGE 00664000
 & DISTAUTO = ''
 THEN EXEC(CMD('EVEEIMON 'MESSAGE) ROUTE(ALL *))) 00665000
 DISPLAY(Y) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y) 00666000
 CONTINUE(Y); 00667000
* 00662000
* SUBSYSTEM IN CTLDOWN STATUS NOT STARTED 00663000
* WARNING ----POSSIBLE CONFLICT WITH EXISTING MESSAGE TABLE STMTS
*
IF MSGID='AOF313I' & TEXT = . 'CTLDOWN' . & TEXT=MESSAGE 00664000
 THEN EXEC(CMD('EVEEIMON 'MESSAGE) ROUTE(ALL *))) 00665000
 DISPLAY(Y) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y) 00666000
 CONTINUE(Y); 00668000
* 00704000
* 00705000
* CURRENT FOCAL POINT IS NEWFP WAS OLDFP FOR THISDOM FOR SYSTEM DDFTREE
* WARNING ----POSSIBLE CONFLICT WITH EXISTING MESSAGE TABLE STMTS 00706000
IF MSGID='AOF660I' & TEXT= MESSAGE & HDRMTYPE--='E' 00707000
 THEN EXEC(CMD('EVEEX041 ' MESSAGE) ROUTE(ALL *))) 00708000
 DISPLAY(Y) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(N) CONTINUE(Y); 00709000
* 00716000
* DOMAIN THISDOM HAS STOPPED COMMUNICATING WITH LOSTDOM 00717000
* WARNING ----POSSIBLE CONFLICT WITH EXISTING MESSAGE TABLE STMTS 00718000

```

```

IF MSGID='AOF662I' & TEXT= MESSAGE & HDRMTYPE~='E' 00719000
 THEN EXEC(CMD('EVEEX040 ' MESSAGE) ROUTE(ALL *)) 00720000
 DISPLAY(Y) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(N) CONTINUE(Y); 00721000
* 00710000
* NO FOCAL POINTS AAVAILABLE * @02 * 00717000
* WARNING ----POSSIBLE CONFLICT WITH EXISTING MESSAGE TABLE STMTS 00718000
IF MSGID='AOF664I' & TEXT= MESSAGE & HDRMTYPE~='E' 00719000
 THEN EXEC(CMD('EVEEX042 ' MESSAGE) ROUTE(ALL *)) 00720000
 DISPLAY(Y) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(N) CONTINUE(Y); 00721000
* 00657000
* -----* 00658000
* CICS STARTUP CONTROL - NO MSG ID ... SIT OVERRIDE REPLY MSG * 00659000
* -----* 00660000
* 00148000
* CICS STARTUP SPECIFY ALTERNATIVE SIT PARMS 00149000
* WARNING ----SPECIFIC AUTOMATED OPERATOR NAME ON ROUTE FUNCTION 00150000
IF HDRMTYPE='>' & 00151000
 TEXT = . 'AND ENTER .END WHEN' . 00152000
 THEN EXEC(CMD('EVESROUT EVEEI115') ROUTE(ONE AUTCICS)) 00153000
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00154000
* ARM STARTED JOB
IF MSGID='IXC812I' & TEXT = MESSAGE
 THEN EXEC(CMD('EVEEARMM ' MESSAGE)
 ROUTE(ONE *))
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y);

```

## EVESCMT2—Message Exit Table for CICS Version 2

|                                                              |           |
|--------------------------------------------------------------|-----------|
| EVESCMT2 TITLE '*** CAO *** XTDCOUT MESSAGE TABLE - PROLOGUE | *00010000 |
| ***'                                                         | 00020000  |
| SPACE 1                                                      | 00030000  |
| EVESCMT2 CSECT ,                                             | 00040000  |
| SPACE 1                                                      | 00050000  |
| *****                                                        | 00060000  |
| *                                                            | 00070000  |
| *                                                            | 00080000  |
| * NAME:                                                      | 00090000  |
| * EVESCMT2                                                   | 00100000  |
| *                                                            | 00110000  |
| *                                                            | 00120000  |
| * TITLE:                                                     | 00130000  |
| * XTDCOUT MESSAGE TABLE                                      | 00140000  |
| *                                                            | 00150000  |
| *                                                            | 00160000  |
| * COMPONENT:                                                 | 00170000  |
| *                                                            | 00180000  |
| * CICS AUTOMATION OPTION (CAO)                               | 00190000  |
| *                                                            | 00200000  |
| * COPYRIGHT=                                                 | 00210000  |
| * LICENSED MATERIALS - PROPERTY OF IBM                       | 00220000  |
| * RESTRICTED MATERIALS OF IBM                                | 00230000  |
| * 5685-151                                                   | 00240000  |
| * (C) COPYRIGHT IBM CORP. 1990, 1995                         | 00250000  |
| *                                                            | 00260000  |
| * US GOVERNMENT USERS RESTRICTED RIGHTS -                    | 00270000  |
| * USE, DUPLICATION OR DISCLOSURE RESTRICTED BY               | 00280000  |
| * GSA ADP SCHEDULE CONTRACT WITH IBM CORP.                   | 00290000  |
| *                                                            | 00300000  |
| *                                                            | 00310000  |
| * ENVIRONMENT:                                               | 00320000  |
| *                                                            | 00330000  |
| * CICS AO/MVS RELEASE 1.1                                    | 00340000  |
| * CICS/MVS RELEASE 2.1.2                                     | 00350000  |
| *                                                            | 00360000  |
| *                                                            | 00370000  |
| * FUNCTION:                                                  | 00380000  |
| *                                                            | 00390000  |
| * TO GENERATE THE XTDCOUT MESSAGE TABLE WHICH IS             | 00400000  |
| * USED BY THE XTDCOUT EXIT PROGRAM EVESCM2 TO                | 00410000  |
| * WRITE SELECTED MESSAGES TO THE MVS CONSOLE.                | 00420000  |
| * THE TABLE IS GENERATED BY SPECIFYING 0 OR A                | 00430000  |
| * NUMBER OF EVEMCMTE TYPE=ENTRY STATEMENTS IN                | 00440000  |
| * BETWEEN AN EVEMCMTE TYPE=INITIAL STATEMENT AND             | 00450000  |
| * AN EVEMCMTE TYPE=FINAL STATEMENT.                          | 00460000  |
| *                                                            | 00470000  |
| *                                                            | 00480000  |
| * ENTRY POINTS:                                              | 00490000  |
| *                                                            | 00500000  |
| * EVESCMT2 AND EVELCMT2                                      | 00510000  |
| *                                                            | 00520000  |
| *                                                            | 00530000  |
| * INVOCATION:                                                | 00540000  |
| *                                                            | 00550000  |
| * N/A                                                        | 00560000  |
| *                                                            | 00570000  |
| *                                                            | 00580000  |
| * INPUT:                                                     | 00590000  |
| *                                                            | 00600000  |
| * N/A                                                        | 00610000  |
| * EJECT                                                      | 00620000  |
| *                                                            | 00630000  |
| *                                                            | 00640000  |
| * PROCESS:                                                   | 00650000  |
| *                                                            | 00660000  |
| * N/A                                                        | 00670000  |
| *                                                            | 00680000  |
| *                                                            | 00690000  |

```

* OUTPUT: * 00700000
* * 00710000
* NONE, THERE IS NO EXECUTABLE CODE. * 00720000
* * 00730000
* * 00740000
* ERRORS DETECTED: * 00750000
* * 00760000
* THE FOLLOWING MNOTES FROM EVEMCMTE MAY BE ISSUED: * 00770000
* * 00780000
* 8,'INCORRECT TYPE SPECIFICATION' * 00790000
* 8,'SEQUENCE ERROR IN EVEMCMTE SPECIFICATIONS' * 00800000
* 8,'REQUIRED PARAMETER(S) OMITTED FOR TYPE=ENTRY' * 00810000
* 8,'INCORRECT QUEUE OR MSGID SPECIFICATION' * 00820000
* 8,'TOO MANY DIFFERENT QUEUES SPECIFIED' * 00830000
* 8,'TOO MANY MESSAGE IDENTIFIERS FOR QUEUE * 00840000
* SPECIFIED' * 00850000
* 8,'TABLE NOT GENERATED DUE TO PREVIOUS ERRORS' * 00860000
* 4,'DUPLICATE QUEUE-MSGID PAIR IGNORED' * 00870000
* 0,'QUEUE AND MSGID ARE IGNORED FOR TYPE=DSECT' * 00880000
* 0,'QUEUE AND MSGID ARE IGNORED FOR TYPE=INITIAL' * 00890000
* 0,'QUEUE AND MSGID ARE IGNORED FOR TYPE=FINAL' * 00900000
* * 00910000
* * 00920000
* EXITS * 00930000
* * 00940000
* N/A * 00950000
* * 00960000
* * 00970000
* REGISTER USAGE: * 00980000
* * 00990000
* N/A * 01000000
* * 01010000
* * 01020000
* NOTES * 01030000
* * 01040000
* ATTRIBUTES: * 01050000
* N/A * 01060000
* RESTRICTIONS: * 01070000
* DEPENDENCIES: * 01080000
* NONE * 01090000
* OTHER: * 01100000
* EVESCMT2 IS TO BE LINK-EDITED TOGETHER WITH * 01110000
* THE EVESCM2 EXIT PROGRAM IN ONE OF THE CICS * 01120000
* DFHRPL LIBRARIES. * 01130000
* EJECT * 01140000
* * 01150000
* * 01160000
* AUTHOR: * 01170000
* * 01180000
* LDR * 01190000
* * 01200000
* * 01210000
* DATE: * 01220000
* * 01230000
* 12/17/90 * 01240000
* * 01250000
* * 01260000
***** 01270000
EJECT 01280000
SPACE 1 01290000
***** 01300000
* APAR# DATE * 01310000
* ----- * 01320000
* IW00027 06/04/91 MISSING SUFFIX ON DFH3463 MSG * 01330000
***** 01340000
TITLE '*** CAO *** XTDCOUT MESSAGE TABLE - GENERATE TABLE *01350000
**' 01360000
***** 01370000
***** 01380000
* * 01390000
* SPECIFY DEFINITIONS FOR XTDCOUT MESSAGE TABLE * 01400000
* * 01410000

```



```


SPACE 2

*
* INITIAL EVEMCMTE SPECIFICATION
*
* 'INITIAL' MUST BE THE FIRST EVEMCMTE TYPE SPECIFIED.
*

SPACE 1
EVEMCMTE TYPE=INITIAL INITIAL ENTRY
SPACE 3
*
* THE EVEMCMTE SPECIFICATIONS BELOW CONTAIN THE MESSAGES
* SENT TO CICS TD QUEUES FOR WHICH CICS AO/MVS IS
* PREPARED TO TAKE ACTION. IF THE ACTION IS NOT REQUIRED
* OR UNDESIRE, THE EVEMCMTE ENTRY SHOULD BE DELETED OR
* COMMENTED OUT.
SPACE 3
W A R N I N G !!!
*
* THE QUEUE SPECIFICATIONS ON THE EVEMCMTE MACROS BELOW
* MUST CONTAIN THE I N D I R E C T QUEUE NAME FOR THE
* CADL, CSMT, AND CSTL DESTINATIONS. CHECK YOUR DCT TO
* ENSURE THIS IS THE CASE. ALTER YOUR DCT OR THE QUEUE
* SPECIFICATIONS BELOW IF REQUIRED!
*
* FOR EXAMPLE IF
*
* DFHDCT TYPE=INDIRECT,
* DESTID=CSMT,
* INDEST=CSSL
*
* IS SPECIFIED, CSSL MUST BE CODED ON THE QUEUE=
* SPECIFICATION FOR ALL CSMT MESSAGES.
EJECT
SPACE 1

*
* DEFINE QUEUES AND MESSAGE IDENTIFIERS
*
* SPECIFY FOR A CICS TD QUEUE THE MESSAGE IDENTIFIER
* OF THE MESSAGE THAT IS TO BE WRITTEN TO THE MVS
* CONSOLE. THE CICS TD QUEUE NAME CAN BE FROM 1 TO
* 4 CHARACTERS. THE MESSAGE IDENTIFIER CAN BE FROM
* 1 TO 16 CHARACTERS. WILD CARD SPECIFICATION (ID*)
* IS ALLOWED.
*

SPACE 1

*
* VTAM ACB MESSAGES
*

SPACE 1
EVEMCMTE TYPE=ENTRY, DEFINE A MESSAGE ENTRY
 QUEUE=CSTL, CICS TD INDIRECT QUEUE NAME
 MSGID=DFH3463* MESSAGE IDENTIFIER
SPACE 2

*
* PROGRAM AND TRANSACTION ABEND MESSAGES
*

SPACE 1
EVEMCMTE TYPE=ENTRY, DEFINE A MESSAGE ENTRY
 QUEUE=CSMT, CICS TD INDIRECT QUEUE NAME
 MSGID=DFH223* MESSAGE IDENTIFIER
SPACE 1

```

```

01420000
01430000
01440000
01450000
01460000
01470000
01480000
01490000
01500000
01510000
01520000
01530000
01540000
01550000
01560000
01570000
01580000
01590000
01600000
01610000
01620000
01630000
01640000
01650000
01660000
01670000
01680000
01690000
01700000
01710000
01720000
01730000
01740000
01750000
01760000
01770000
01780000
01790000
01800000
01810000
01820000
01830000
01840000
01850000
01860000
01870000
01880000
01890000
01900000
01910000
01920000
01930000
01940000
01950000
01960000
01970000
01980000
01990000
*02000000
*02010000
02020000
02030000
02040000
02050000
02060000
02070000
02080000
02090000
*02100000
*02110000
02120000
02130000

```

|                                                       |                             |           |
|-------------------------------------------------------|-----------------------------|-----------|
| EVEMCMTE TYPE=ENTRY,                                  | DEFINE A MESSAGE ENTRY      | *02140000 |
| QUEUE=CSMT,                                           | CICS TD INDIRECT QUEUE NAME | *02150000 |
| MSGID=DFH224*                                         | MESSAGE IDENTIFIER          | 02160000  |
| SPACE 2                                               |                             | 02170000  |
| *****                                                 |                             | 02180000  |
| *                                                     |                             | 02190000  |
| * AUTOINSTALL ERROR MESSAGES *                        |                             | 02200000  |
| *                                                     |                             | 02210000  |
| *****                                                 |                             | 02220000  |
| SPACE 1                                               |                             | 02230000  |
| EVEMCMTE TYPE=ENTRY,                                  | DEFINE A MESSAGE ENTRY      | *02240000 |
| QUEUE=CSMT,                                           | CICS TD INDIRECT QUEUE NAME | *02250000 |
| MSGID=DFH3482                                         | MESSAGE IDENTIFIER          | 02260000  |
| SPACE 1                                               |                             | 02270000  |
| EVEMCMTE TYPE=ENTRY,                                  | DEFINE A MESSAGE ENTRY      | *02280000 |
| QUEUE=CSMT,                                           | CICS TD INDIRECT QUEUE NAME | *02290000 |
| MSGID=DFH3483                                         | MESSAGE IDENTIFIER          | 02300000  |
| SPACE 1                                               |                             | 02310000  |
| EVEMCMTE TYPE=ENTRY,                                  | DEFINE A MESSAGE ENTRY      | *02320000 |
| QUEUE=CADL,                                           | CICS TD INDIRECT QUEUE NAME | *02330000 |
| MSGID=DFH5935I                                        | MESSAGE IDENTIFIER          | 02340000  |
| SPACE 1                                               |                             | 02350000  |
| EVEMCMTE TYPE=ENTRY,                                  | DEFINE A MESSAGE ENTRY      | *02360000 |
| QUEUE=CADL,                                           | CICS TD INDIRECT QUEUE NAME | *02370000 |
| MSGID=DFH5966I                                        | MESSAGE IDENTIFIER          | 02380000  |
| EJECT                                                 |                             | 02390000  |
| SPACE 1                                               |                             | 02400000  |
| *****                                                 |                             | 02410000  |
| *                                                     |                             | 02420000  |
| * 'FINAL' MUST BE THE LAST EVEMCMTE TYPE SPECIFIED. * |                             | 02430000  |
| *                                                     |                             | 02440000  |
| *****                                                 |                             | 02450000  |
| SPACE 1                                               |                             | 02460000  |
| EVEMCMTE TYPE=FINAL                                   | REQUIRED END                | 02470000  |
| TITLE '*** CAO *** XTDCOUT MESSAGE TABLE - PROLOGUE   | **'                         | *02480000 |
|                                                       |                             | 02490000  |
| SPACE 1                                               |                             | 02500000  |
| EVEMCMTE TYPE=DSECT                                   |                             | 02510000  |
| EJECT                                                 |                             | 02520000  |
| SPACE 1                                               |                             | 02530000  |
| END EVESCMT2                                          |                             | 02540000  |

## EVESCMT3—Message Exit Table for CICS Version 3 and Higher

|                                                             |           |
|-------------------------------------------------------------|-----------|
| EVESCMT3 TITLE '*** CAO *** XMEOUT MESSAGE TABLE - PROLOGUE | *00010000 |
| ***'                                                        | 00020000  |
| SPACE 1                                                     | 00030000  |
| EVESCMT3 CSECT , ESTABLISH CONTROL SECTION                  | 00040000  |
| SPACE 1                                                     | 00050000  |
| *****                                                       | 00060000  |
| *                                                           | 00070000  |
| *                                                           | 00080000  |
| * NAME:                                                     | 00090000  |
| * EVESCMT3                                                  | 00100000  |
| *                                                           | 00110000  |
| *                                                           | 00120000  |
| * TITLE:                                                    | 00130000  |
| * XMEOUT MESSAGE TABLE                                      | 00140000  |
| *                                                           | 00150000  |
| *                                                           | 00160000  |
| * COMPONENT:                                                | 00170000  |
| *                                                           | 00180000  |
| * CICS AUTOMATION OPTION (CAO)                              | 00190000  |
| *                                                           | 00200000  |
| * COPYRIGHT=                                                | 00210000  |
| * LICENSED MATERIALS - PROPERTY OF IBM                      | 00220000  |
| * RESTRICTED MATERIALS OF IBM                               | 00230000  |
| * 5685-151                                                  | 00240000  |
| * (C) COPYRIGHT IBM CORP. 1990, 1994                        | 00250000  |
| *                                                           | 00260000  |
| * US GOVERNMENT USERS RESTRICTED RIGHTS -                   | 00270000  |
| * USE, DUPLICATION OR DISCLOSURE RESTRICTED BY              | 00280000  |
| * GSA ADP SCHEDULE CONTRACT WITH IBM CORP.                  | 00290000  |
| *                                                           | 00300000  |
| *                                                           | 00310000  |
| * ENVIRONMENT:                                              | 00320000  |
| *                                                           | 00330000  |
| * CAO RELEASE 1.0                                           | 00340000  |
| * CICS/ESA RELEASE 3.2                                      | 00350000  |
| *                                                           | 00360000  |
| *                                                           | 00370000  |
| * FUNCTION:                                                 | 00380000  |
| *                                                           | 00390000  |
| * TO GENERATE THE XMEOUT MESSAGE TABLE WHICH IS             | 00400000  |
| * USED BY THE XMEOUT EXIT PROGRAM EVESCM3 TO                | 00410000  |
| * ROUTE SELECTED MESSAGES TO THE MVS CONSOLE.               | 00420000  |
| * THE TABLE IS GENERATED BY SPECIFYING 0 OR A               | 00430000  |
| * NUMBER OF EVEMXMET TYPE=ENTRY STATEMENTS IN               | 00440000  |
| * BETWEEN AN EVEMXMET TYPE=INITIAL STATEMENT AND            | 00450000  |
| * AN EVEMXMET TYPE=FINAL STATEMENT.                         | 00460000  |
| *                                                           | 00470000  |
| *                                                           | 00480000  |
| * ENTRY POINTS:                                             | 00490000  |
| *                                                           | 00500000  |
| * EVESCMT3 AND EVELCMT3                                     | 00510000  |
| *                                                           | 00520000  |
| *                                                           | 00530000  |
| * INVOCATION:                                               | 00540000  |
| *                                                           | 00550000  |
| * N/A                                                       | 00560000  |
| *                                                           | 00570000  |
| *                                                           | 00580000  |
| * INPUT:                                                    | 00590000  |

|   |                                                   |   |          |
|---|---------------------------------------------------|---|----------|
| * |                                                   | * | 00600000 |
| * | N/A                                               | * | 00610000 |
| * | EJECT                                             | * | 00620000 |
| * |                                                   | * | 00630000 |
| * |                                                   | * | 00640000 |
| * | PROCESS:                                          | * | 00650000 |
| * |                                                   | * | 00660000 |
| * | N/A                                               | * | 00670000 |
| * |                                                   | * | 00680000 |
| * |                                                   | * | 00690000 |
| * | OUTPUT:                                           | * | 00700000 |
| * |                                                   | * | 00710000 |
| * | NONE, THERE IS NO EXECUTABLE CODE.                | * | 00720000 |
| * |                                                   | * | 00730000 |
| * |                                                   | * | 00740000 |
| * | ERRORS DETECTED:                                  | * | 00750000 |
| * |                                                   | * | 00760000 |
| * | THE FOLLOWING MNOTES FROM EVEMXMET MAY BE ISSUED: | * | 00770000 |
| * |                                                   | * | 00780000 |
| * | 8,'INCORRECT TYPE SPECIFICATION'                  | * | 00790000 |
| * | 8,'SEQUENCE ERROR IN EVEMXMET SPECIFICATIONS'     | * | 00800000 |
| * | 8,'REQUIRED PARAMETER(S) OMITTED FOR TYPE=ENTRY'  | * | 00810000 |
| * | 8,'INCORRECT QUEUE OR MSGID SPECIFICATION'        | * | 00820000 |
| * | 8,'TOO MANY DIFFERENT QUEUES SPECIFIED'           | * | 00830000 |
| * | 8,'TOO MANY MESSAGE IDENTIFIERS FOR QUEUE         | * | 00840000 |
| * | SPECIFIED'                                        | * | 00850000 |
| * | 8,'TABLE NOT GENERATED DUE TO PREVIOUS ERRORS'    | * | 00860000 |
| * | 4,'DUPLICATE QUEUE-MSGID PAIR IGNORED'            | * | 00870000 |
| * | 0,'QUEUE AND MSGID ARE IGNORED FOR TYPE=DSECT'    | * | 00880000 |
| * | 0,'QUEUE AND MSGID ARE IGNORED FOR TYPE=INITIAL'  | * | 00890000 |
| * | 0,'QUEUE AND MSGID ARE IGNORED FOR TYPE=FINAL'    | * | 00900000 |
| * |                                                   | * | 00910000 |
| * |                                                   | * | 00920000 |
| * | EXITS                                             | * | 00930000 |
| * |                                                   | * | 00940000 |
| * | N/A                                               | * | 00950000 |
| * |                                                   | * | 00960000 |
| * |                                                   | * | 00970000 |
| * | REGISTER USAGE:                                   | * | 00980000 |
| * |                                                   | * | 00990000 |
| * | N/A                                               | * | 01000000 |
| * |                                                   | * | 01010000 |
| * |                                                   | * | 01020000 |
| * | NOTES                                             | * | 01030000 |
| * |                                                   | * | 01040000 |
| * | ATTRIBUTES:                                       | * | 01050000 |
| * | N/A                                               | * | 01060000 |
| * | RESTRICTIONS:                                     | * | 01070000 |
| * | DEPENDENCIES:                                     | * | 01080000 |
| * | NONE                                              | * | 01090000 |
| * | OTHER:                                            | * | 01100000 |
| * | EVESCM3 IS TO BE LINK-EDITED TOGETHER WITH        | * | 01110000 |
| * | THE EVESCM3 EXIT PROGRAM IN ONE OF THE CICS       | * | 01120000 |
| * | DFHRPL LIBRARIES.                                 | * | 01130000 |
| * | EJECT                                             | * | 01140000 |
| * |                                                   | * | 01150000 |
| * |                                                   | * | 01160000 |
| * | AUTHOR:                                           | * | 01170000 |
| * |                                                   | * | 01180000 |
| * | LDR                                               | * | 01190000 |
| * |                                                   | * | 01200000 |
| * |                                                   | * | 01210000 |
| * | DATE:                                             | * | 01220000 |
| * |                                                   | * | 01230000 |

```

* 02/15/91 * 01240000
* * 01250000
* * 01260000
***** * 01270000
 EJECT * 01280000
 SPACE 1 * 01290000
***** * 01300000
* APAR# DATE * 01310000
* $03=OW35605,V1R4,03DEC98,APC(JK): CICS WAVE 2 * 01320000
* * 01320100
* $02=PN41819 06/04/93 GD CICS AO SYS LIB FOR MODULE EVESCM T3 IS LISTED* 01330000
* * 01340000
* BY SMP/E AS SEVEMOD2 AND AEVEMOD2 AFTER
* UN40897 BUT SHOULD BE ONLY SEVEMOD2
* * 01350000
* * 01360000
* $01=PN39560 04/19/93 GD JCLIN ERROR IN UN28322 CAUSES IEBCOPY * 01370000
* * 01380000
* MSGIEB102I
* * 01390000
* * 01400000
***** * 01410000
 TITLE '*** CAO *** XMEOUT MESSAGE TABLE - GENERATE TABLE * 01420000
 ** '
***** * 01430000
***** * 01440000
***** * 01450000
* * 01460000
* SPECIFY DEFINITIONS FOR XMEOUT MESSAGE TABLE * 01470000
* * 01480000
***** * 01490000
***** * 01500000
 SPACE 2 * 01510000
***** * 01520000
* * 01530000
* INITIAL EVEMXMET SPECIFICATION * 01540000
* * 01550000
* 'INITIAL' MUST BE THE FIRST EVEMXMET TYPE SPECIFIED. * 01560000
* * 01570000
***** * 01580000
 SPACE 1 * 01590000
 EVEMXMET TYPE=INITIAL INITIAL ENTRY * 01600000
 SPACE 3 * 01610000
* THE EVEMXMET SPECIFICATIONS BELOW CONTAIN THE MESSAGES * 01620000
* SENT TO CICS TD QUEUES FOR WHICH CICS AO/MVS IS * 01630000
* PREPARED TO TAKE ACTION. IF THE ACTION IS NOT REQUIRED * 01640000
* OR UNDESIRE D, THE EVEMXMET ENTRY SHOULD BE DELETED OR * 01650000
* COMMENTED OUT. * 01660000
 SPACE 3 * 01670000
* W A R N I N G !!! * 01680000
* * 01690000
* THE QUEUE SPECIFICATIONS ON THE EVEMXMET MACROS BELOW * 01700000
* MUST CONTAIN THE A C T U A L CICS TD QUEUE NAME. * 01710000
* * 01720000
* FOR EXAMPLE IF * 01730000
* * 01740000
* DFHDCT TYPE=INDIRECT, * 01750000
* DESTID=CSMT, * 01760000
* INDEST=CSSL * 01770000
* * 01780000
* IS SPECIFIED, CSMT MUST BE CODED ON THE QUEUE= * 01790000
* SPECIFICATION FOR ALL CSMT MESSAGES. * 01800000
 EJECT * 01810000
 SPACE 1 * 01820000
***** * 01830000
***** * 01840000
* * 01850000
* DEFINE QUEUES AND MESSAGE IDENTIFIERS * 01860000

```

|       |                                                    |        |           |
|-------|----------------------------------------------------|--------|-----------|
| *     |                                                    | *      | 01870000  |
| *     | SPECIFY FOR A CICS TD QUEUE THE MESSAGE IDENTIFIER | *      | 01880000  |
| *     | OF THE MESSAGE THAT IS TO BE WRITTEN TO THE MVS    | *      | 01890000  |
| *     | CONSOLE. THE CICS TD QUEUE NAME MUST HAVE A LENGTH | *      | 01900000  |
| *     | FROM 1 TO 4 CHARACTERS. THE MESSAGE IDENTIFIER     | *      | 01910000  |
| *     | MUST HAVE THE FORMAT: DFHDINNNN                    | *      | 01920000  |
| *     | WHERE: DI IS THE DOMAIN IDENTIFIER                 | *      | 01930000  |
| *     | NNNN IS THE MESSAGE NUMBER                         | *      | 01940000  |
| *     |                                                    | *      | 01950000  |
| ***** |                                                    |        | 01960000  |
| ***** |                                                    |        | 01970000  |
|       | SPACE 1                                            |        | 01980000  |
| ***** |                                                    |        | 01990000  |
| *     |                                                    | *      | 02000000  |
| *     | VTAM ACB MESSAGES                                  | *      | 02010000  |
| *     |                                                    | *      | 02020000  |
| ***** |                                                    |        | 02030000  |
|       | SPACE 1                                            |        | 02040000  |
|       | EVEXMET TYPE=ENTRY, DEFINE A MESSAGE ENTRY         | *      | *02050000 |
|       | QUEUE=CSNE, ACTUAL CICS TD QUEUE NAME              | *      | *02060000 |
|       | MSGID=DFHZC3463 MESSAGE IDENTIFIER                 |        | 02070000  |
|       | SPACE 2                                            |        | 02080000  |
| ***** |                                                    |        | 02090000  |
| *     |                                                    | *      | 02100000  |
| *     | PROGRAM AND TRANSACTION ABEND MESSAGES             | *      | 02110000  |
| *     |                                                    | *      | 02120000  |
| ***** |                                                    |        | 02130000  |
|       | SPACE 1                                            |        | 02140000  |
|       | EVEXMET TYPE=ENTRY, DEFINE A MESSAGE ENTRY         | *      | *02150000 |
|       | QUEUE=CSMT, ACTUAL CICS TD QUEUE NAME              | *      | *02160000 |
|       | MSGID=DFHAC2230 MESSAGE IDENTIFIER                 |        | 02170000  |
|       | SPACE 1                                            |        | 02170100  |
|       | EVEXMET TYPE=ENTRY, DEFINE A MESSAGE ENTRY         | 3@03A* | *02170200 |
|       | QUEUE=CSMT, ACTUAL CICS TD QUEUE NAME              | *      | *02170300 |
|       | MSGID=DFHAC2231 MESSAGE IDENTIFIER                 |        | 02170400  |
|       | SPACE 1                                            |        | 02170500  |
|       | EVEXMET TYPE=ENTRY, DEFINE A MESSAGE ENTRY         | 3@03A* | *02170600 |
|       | QUEUE=CSMT, ACTUAL CICS TD QUEUE NAME              | *      | *02170700 |
|       | MSGID=DFHAC2232 MESSAGE IDENTIFIER                 |        | 02170800  |
|       | SPACE 1                                            |        | 02170900  |
|       | EVEXMET TYPE=ENTRY, DEFINE A MESSAGE ENTRY         | 3@03A* | *02171000 |
|       | QUEUE=CSMT, ACTUAL CICS TD QUEUE NAME              | *      | *02171100 |
|       | MSGID=DFHAC2233 MESSAGE IDENTIFIER                 |        | 02171200  |
|       | SPACE 1                                            |        | 02180000  |
|       | EVEXMET TYPE=ENTRY, DEFINE A MESSAGE ENTRY         | *      | *02190000 |
|       | QUEUE=CSMT, ACTUAL CICS TD QUEUE NAME              | *      | *02200000 |
|       | MSGID=DFHAC2236 MESSAGE IDENTIFIER                 |        | 02210000  |
|       | SPACE 1                                            |        | 02220000  |
|       | EVEXMET TYPE=ENTRY, DEFINE A MESSAGE ENTRY         | *      | *02230000 |
|       | QUEUE=CSMT, ACTUAL CICS TD QUEUE NAME              | *      | *02240000 |
|       | MSGID=DFHAC2237 MESSAGE IDENTIFIER                 |        | 02250000  |
|       | SPACE 1                                            |        | 02260000  |
|       | EVEXMET TYPE=ENTRY, DEFINE A MESSAGE ENTRY         | *      | *02270000 |
|       | QUEUE=CSMT, ACTUAL CICS TD QUEUE NAME              | *      | *02280000 |
|       | MSGID=DFHAC2238 MESSAGE IDENTIFIER                 |        | 02290000  |
|       | SPACE 1                                            |        | 02300000  |
|       | EVEXMET TYPE=ENTRY, DEFINE A MESSAGE ENTRY         | *      | *02310000 |
|       | QUEUE=CSMT, ACTUAL CICS TD QUEUE NAME              | *      | *02320000 |
|       | MSGID=DFHAC2240 MESSAGE IDENTIFIER                 |        | 02330000  |
|       | SPACE 1                                            |        | 02340000  |
|       | EVEXMET TYPE=ENTRY, DEFINE A MESSAGE ENTRY         | *      | *02350000 |
|       | QUEUE=CSMT, ACTUAL CICS TD QUEUE NAME              | *      | *02360000 |
|       | MSGID=DFHAC2241 MESSAGE IDENTIFIER                 |        | 02370000  |
|       | SPACE 1                                            |        | 02380000  |

|                              |                           |                |
|------------------------------|---------------------------|----------------|
| EVEMXMET TYPE=ENTRY,         | DEFINE A MESSAGE ENTRY    | *02390000      |
| QUEUE=CSMT,                  | ACTUAL CICS TD QUEUE NAME | *02400000      |
| MSGID=DFHAC2242              | MESSAGE IDENTIFIER        | 02410000       |
| SPACE 1                      |                           | 02420000       |
| EVEMXMET TYPE=ENTRY,         | DEFINE A MESSAGE ENTRY    | *02430000      |
| QUEUE=CSMT,                  | ACTUAL CICS TD QUEUE NAME | *02440000      |
| MSGID=DFHAC2243              | MESSAGE IDENTIFIER        | 02450000       |
| SPACE 1                      |                           | 02450100       |
| EVEMXMET TYPE=ENTRY,         | DEFINE A MESSAGE ENTRY    | 3@03A*02450200 |
| QUEUE=CSMT,                  | ACTUAL CICS TD QUEUE NAME | *02450300      |
| MSGID=DFHAC2245              | MESSAGE IDENTIFIER        | 02450400       |
| SPACE 1                      |                           | 02450500       |
| EVEMXMET TYPE=ENTRY,         | DEFINE A MESSAGE ENTRY    | 3@03A*02450600 |
| QUEUE=CSMT,                  | ACTUAL CICS TD QUEUE NAME | *02450700      |
| MSGID=DFHAC2246              | MESSAGE IDENTIFIER        | 02450800       |
| SPACE 1                      |                           | 02450900       |
| EVEMXMET TYPE=ENTRY,         | DEFINE A MESSAGE ENTRY    | 3@03A*02451000 |
| QUEUE=CSMT,                  | ACTUAL CICS TD QUEUE NAME | *02451100      |
| MSGID=DFHAC2247              | MESSAGE IDENTIFIER        | 02451200       |
| SPACE 1                      |                           | 02451300       |
| EVEMXMET TYPE=ENTRY,         | DEFINE A MESSAGE ENTRY    | 3@03A*02451400 |
| QUEUE=CSMT,                  | ACTUAL CICS TD QUEUE NAME | *02451500      |
| MSGID=DFHAC2248              | MESSAGE IDENTIFIER        | 02451600       |
| SPACE 1                      |                           | 02451700       |
| EVEMXMET TYPE=ENTRY,         | DEFINE A MESSAGE ENTRY    | 3@03A*02451800 |
| QUEUE=CSMT,                  | ACTUAL CICS TD QUEUE NAME | *02451900      |
| MSGID=DFHAC2249              | MESSAGE IDENTIFIER        | 02452000       |
| SPACE 1                      |                           | 02452100       |
| EVEMXMET TYPE=ENTRY,         | DEFINE A MESSAGE ENTRY    | 3@03A*02452200 |
| QUEUE=CSMT,                  | ACTUAL CICS TD QUEUE NAME | *02452300      |
| MSGID=DFHAC2250              | MESSAGE IDENTIFIER        | 02452400       |
| SPACE 1                      |                           | 02452500       |
| EVEMXMET TYPE=ENTRY,         | DEFINE A MESSAGE ENTRY    | 3@03A*02452600 |
| QUEUE=CSMT,                  | ACTUAL CICS TD QUEUE NAME | *02452700      |
| MSGID=DFHAC2251              | MESSAGE IDENTIFIER        | 02452800       |
| SPACE 1                      |                           | 02452900       |
| EVEMXMET TYPE=ENTRY,         | DEFINE A MESSAGE ENTRY    | 3@03A*02453000 |
| QUEUE=CSMT,                  | ACTUAL CICS TD QUEUE NAME | *02453100      |
| MSGID=DFHAC2252              | MESSAGE IDENTIFIER        | 02453200       |
| SPACE 1                      |                           | 02453300       |
| EVEMXMET TYPE=ENTRY,         | DEFINE A MESSAGE ENTRY    | 3@03A*02453400 |
| QUEUE=CSMT,                  | ACTUAL CICS TD QUEUE NAME | *02453500      |
| MSGID=DFHAC2253              | MESSAGE IDENTIFIER        | 02453600       |
| EJECT                        |                           | 02460000       |
| SPACE 1                      |                           | 02470000       |
| *****                        |                           | 02480000       |
| *                            |                           | 02490000       |
| * AUTOINSTALL ERROR MESSAGES |                           | 02500000       |
| *                            |                           | 02510000       |
| *****                        |                           | 02520000       |
| SPACE 1                      |                           | 02530000       |
| EVEMXMET TYPE=ENTRY,         | DEFINE A MESSAGE ENTRY    | *02540000      |
| QUEUE=CSNE,                  | ACTUAL CICS TD QUEUE NAME | *02550000      |
| MSGID=DFHZC3482              | MESSAGE IDENTIFIER        | 02560000       |
| SPACE 1                      |                           | 02570000       |
| EVEMXMET TYPE=ENTRY,         | DEFINE A MESSAGE ENTRY    | *02580000      |
| QUEUE=CSNE,                  | ACTUAL CICS TD QUEUE NAME | *02590000      |
| MSGID=DFHZC3483              | MESSAGE IDENTIFIER        | 02600000       |
| SPACE 1                      |                           | 02610000       |
| EVEMXMET TYPE=ENTRY,         | DEFINE A MESSAGE ENTRY    | *02620000      |
| QUEUE=CADL,                  | ACTUAL CICS TD QUEUE NAME | *02630000      |
| MSGID=DFHZC5966              | MESSAGE IDENTIFIER        | 02640000       |
| EJECT                        |                           | 02650000       |

|                                                          |           |
|----------------------------------------------------------|-----------|
| SPACE 1                                                  | 02660000  |
| *****                                                    | 02670000  |
| *                                                        | 02680000  |
| * 'FINAL' MUST BE THE LAST EVEMXMET TYPE SPECIFIED.      | 02690000  |
| *                                                        | 02700000  |
| *****                                                    | 02710000  |
| SPACE 1                                                  | 02720000  |
| EVEMXMET TYPE=FINAL REQUIRED END                         | 02730000  |
| TITLE '*** CAO *** XMEOUT MESSAGE TABLE - DUMMY SECTIONS | *02740000 |
| ***'                                                     | 02750000  |
| SPACE 1                                                  | 02760000  |
| EVEMXMET TYPE=DSECT                                      | 02770000  |
| EJECT                                                    | 02780000  |
| SPACE 1                                                  | 02790000  |
| END EVESCMT3                                             | 02800000  |



## EVECMD—NetView Command Model Statements Member CICS Automation Entries

```

* (C) COPYRIGHT IBM CORP. 1990,1995,1998 - ALL RIGHTS RESERVED *
* * *
* LICENSED MATERIALS - PROPERTY OF IBM *
* RESTRICTED MATERIALS OF IBM *
* 5685-151 *
* *
* US GOVERNMENT USERS RESTRICTED RIGHTS - *
* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY *
* GSA ADP SCHEDULE CONTRACT WITH IBM CORP. *

* APAR# DATE *
* ----- *
* $01=OW35605,V1R4,22OCT98,APC(JK): CICS WAVE 2 *
* $L1=ARM 09/30/95 PJ CHANGES FOR AOC/CICS 1.4 *
* 06/06/94 MP CHANGES FOR AOC/CICS 1.3 *
* $01=PN37409 04/27/93 BM MSGEVE004I SECURITY CHECK KEY VALUE=STARTUP*
* *
* PN04832 09/18/91 TP CICS DLY NOT WORKING DUE TO MISS-SPELLING AS*
* CICS DLY IN EVECMD. CICS AO PROGRAMMING *
* MANUAL SHOWED THIS CMD AS CICS DLY ON P.99. *
* *
* IW00019 06/05/91 ADD EVESNACK FOR REMOTE DOMAIN RESPONSES *

* START OF CICS AUTOMATION OPTION COMMAND PROCESSOR DEFINITIONS *

EVESTS CMDMDL MOD=EVE SCSTS,TYPE=R,RES=Y,ECHO=Y
EVEASTS CMDMDL MOD=EVE SMSTS,TYPE=D,RES=Y,PARSE=N
* CICS MAIN MENU PANEL
EVEE0000 CMDMDL MOD=DSICCP
CMD SYN CICS
* CICS TRACE FUNCTION
EVESTRAC CMDMDL MOD=EVESTRAC,RES=Y,PARSE=Y
CMD SYN EVETRACE
* CICS COMMON FUNCTIONS, LINE MODE COMMANDS
EVEEC011 CMDMDL MOD=DSICCP
CMD SYN CEMTPPI
EVEERINT CMDMDL MOD=DSICCP
CMD SYN CICSRSYC
EVEET015 CMDMDL MOD=DSICCP
CMD SYN CICS SHUT
*
EVEECMSH CMDMDL MOD=DSICCP
CMD SYN CMAS SHUT
EVEEARMW CMDMDL MOD=DSICCP
EVEERLSI CMDMDL MOD=DSICCP
EVEERLSC CMDMDL MOD=DSICCP
EVEETUOW CMDMDL MOD=DSICCP
*
EVEET050 CMDMDL MOD=DSICCP
CMD SYN CICS PURG
EVEEX016 CMDMDL MOD=DSICCP
CMD SYN CICS DLY
EVEEXPST CMDMDL MOD=DSICCP
CMD SYN CICS POST
EVEEX017 CMDMDL MOD=DSICCP
CMD SYN CICSRCMD
EVEEJ202 CMDMDL MOD=DSICCP

```

6@01A

```

 CMDSYN CICSTRCE
EVEEJ402 CMDMDL MOD=DSICCP
 CMDSYN CICSMDHR
EVEEMHCR CMDMDL MOD=DSICCP
 CMDSYN CICSHLTH
EVEEI112 CMDMDL MOD=DSICCP
 CMDSYN CICSSTRT
EVEET112 CMDMDL MOD=DSICCP
 CMDSYN CICSSTOP
EVEEI116 CMDMDL MOD=DSICCP
 CMDSYN CICSVRD
EVEEYLMZ CMDMDL MOD=DSICCP
 CMDSYN CICSLM
* CICS SECURITY CHECKING
EVESX001 CMDMDL MOD=EVESX001,RES=Y,PARSE=Y
 CMDSYN CICSSEC
%INCLUDE EVESECUR
*
* MESSAGE BUILD/ROUTE
EVESX002 CMDMDL MOD=EVESX002,RES=Y,PARSE=Y
 CMDSYN CICSMSG
EVESX003 CMDMDL MOD=EVESX003,RES=Y,PARSE=Y
* TRIGGER/EVENT
EVESX004 CMDMDL MOD=EVESX004,RES=Y,PARSE=Y
* SINGLE POINT OF CONTROL (SPOC)
EVESX022 CMDMDL MOD=EVESX022,RES=Y,PARSE=Y
 CMDSYN CICSQRY
EVESXDMB CMDMDL MOD=EVESXDMB,RES=Y,PARSE=Y
EVESXDMO CMDMDL MOD=EVESXDMO,RES=Y,PARSE=Y
EVESXDMR CMDMDL MOD=EVESXDMR,RES=Y,PARSE=Y
* MESSAGE REROUTING
EVESROUT CMDMDL MOD=EVESROUT,RES=Y
EVESXGBL CMDMDL MOD=EVESXGBL,RES=Y,PARSE=Y
* SERVICE PERIOD
EVESSSRV CMDMDL MOD=EVESSSRV,RES=Y,PARSE=Y
EVESXDAT CMDMDL MOD=EVESXDAT,RES=Y,PARSE=Y
* CONSOLE INTERFACE
EVESNMOD CMDMDL MOD=EVESNMOD,RES=Y,PARSE=Y
EVESX060 CMDMDL MOD=EVESX060,RES=N,PARSE=Y
* PROGRAM-TO-PROGRAM INTERFACE
EVESNCCI CMDMDL MOD=EVESNCCI,TYPE=R,PARSE=N
EVESNRSP CMDMDL MOD=EVESNRSP,TYPE=R,PARSE=N
EVESNBRE CMDMDL MOD=EVESNBRE,TYPE=R,PARSE=N
EVESNACK CMDMDL MOD=EVESNACK,TYPE=R,PARSE=N
* SMF
EVESUSMF CMDMDL MOD=EVESUSMF,TYPE=R,PARSE=Y
* MEMBER READ
EVESY001 CMDMDL MOD=EVESY001,RES=N,PARSE=Y
*
* ADD ALL SUBSYSTEM TABLE MODULE FOR PERFORMANCE @L1
EVEEAAS CMDMDL MOD=DSICCP
*
* END OF CICS ENTRIES

* END OF CICS AUTOMATION OPTION COMMAND PROCESSOR DEFINITIONS *

```

---

## EVESECUR—Security Definitions (Included by EVECMD)

```

* (C) COPYRIGHT IBM CORP. 1994, 1995 - ALL RIGHTS RESERVED *
* *
* LICENSED MATERIALS - PROPERTY OF IBM *
* RESTRICTED MATERIALS OF IBM *
* 5685-151 *
* *
* US GOVERNMENT USERS RESTRICTED RIGHTS - *
* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY *
* GSA ADP SCHEDULE CONTRACT WITH IBM CORP. *

* APAR# DATE *
* ----- *
* *

* DESCRIPTION: CICS SECURITY CHECKING *

* NOTES: *
* KEYCLASS CAN BE AN CICS 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. *
* *

CICST1 KEYCLASS 1 SECURITY CLASS FOR THIS CICS
INQUIRY VALCLASS 1 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
CEMT VALCLASS 2 CEMT INQ/SET PROCESSING
LMT VALCLASS 2 LINK MONITORING PROCESSING
HEALTH VALCLASS 2 HEALTH CHECKING (SAMPLE)
SUPPORT VALCLASS 2 TRACE, MESSAGE FUNCTIONS
EVEPPICT VALCLASS 2 START,STOP PPI INTERFACE
BRDCAST VALCLASS 2 BROADCAST FUNCTION
=OTHER VALCLASS 2 ANY FUNCTION SPECIFIED
*
=OTHER KEYCLASS 9 ANY CICS NOT PREV. DEFINED
=OTHER VALCLASS 9 ANY FUNCTION FOR UNDEF. CICS
```

---

## EVEDMN—NetView Domain Member CICS Automation Entries

```
* *****
* (C) COPYRIGHT IBM CORP. 1990, 1995 - ALL RIGHTS RESERVED
*
* LICENSED MATERIALS - PROPERTY OF IBM
* RESTRICTED MATERIALS OF IBM
* 5685-151
*
* US GOVERNMENT USERS RESTRICTED RIGHTS -
* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY
* GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
* *****
* APAR# DATE *
* ----- -
*
* *****
* ENABLE CICS FEATURE PPI RECEIVER TASKS
*
* TASK MOD=EVESNPPI,TSKID=EVENTASK,MEM=EVENTASK,PRI=5
*
```

---

## EVEOPF—NetView Operator Definitions CICS Automation Entries

```

* (C) COPYRIGHT IBM CORP. 1990, 1995 - ALL RIGHTS RESERVED * 00020025
* * 00020025
* LICENSED MATERIALS - PROPERTY OF IBM * 00020025
* RESTRICTED MATERIALS OF IBM * 00020025
* 5685-151 * 00020025
* * 00020025
* US GOVERNMENT USERS RESTRICTED RIGHTS - * 00020025
* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY * 00020025
* GSA ADP SCHEDULE CONTRACT WITH IBM CORP. * 00020025

* APAR# DATE *
* ----- -
* *
* * 00020025

* ADDITIONAL AUTOTASK IDS FOR CICS ACO OPTION * 00020025

AUTCICS OPERATOR PASSWORD=AUTCICS 00671007
 PROFILE EVEPRFAO 00672007
AUTC01A OPERATOR PASSWORD=AUTC01A 00671007
 PROFILE EVEPRFAO 00672007
AUTC01B OPERATOR PASSWORD=AUTC01B 00671007
 PROFILE EVEPRFAO 00672007
AUTCPPI OPERATOR PASSWORD=AUTCPPI 00671007
 PROFILE EVEPRFAO 00672007
```

---

## EVETREE—The Status Display Facility Tree Structure

```
/* ***** */
/* (C) COPYRIGHT IBM CORP. 1990, 1995 - ALL RIGHTS RESERVED */
/* */
/* LICENSED MATERIALS - PROPERTY OF IBM */
/* RESTRICTED MATERIALS OF IBM */
/* 5685-151 */
/* */
/* US GOVERNMENT USERS RESTRICTED RIGHTS - */
/* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY */
/* GSA ADP SCHEDULE CONTRACT WITH IBM CORP. */
/* ***** */
/* APAR# DATE */
/* ----- ----- */
/* */
/* ***** */
/* CICS DISPLAY FACILITY "TREE" DEFINITIONS */
2 CICS
3 CICSHLTH
3 CICSLMT
3 CICSAUTO
3 CICSMSG
3 CICSSTG
4 CICSSOS
4 CICSVIOL
3 CICSTIMR
3 CICSTRAN
3 VTAMACB
```

---

## EVEPNLS—The Status Display Facility Panels

```
/* ***** */
/* (C) COPYRIGHT IBM CORP. 1990, 1995 - ALL RIGHTS RESERVED */
/* */
/* LICENSED MATERIALS - PROPERTY OF IBM */
/* RESTRICTED MATERIALS OF IBM */
/* 5685-151 */
/* */
/* US GOVERNMENT USERS RESTRICTED RIGHTS - */
/* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY */
/* GSA ADP SCHEDULE CONTRACT WITH IBM CORP. */
/* ***** */
/* APAR# DATE */
/* ----- ----- */
/* */
/* ***** */
%INCLUDE(EVED0001)
%INCLUDE(EVEDACB1)
%INCLUDE(EVEDACB2)
%INCLUDE(EVEDACBA)
%INCLUDE(EVEDACBB)
%INCLUDE(EVEDAUI1)
%INCLUDE(EVEDAUI2)
%INCLUDE(EVEDAUIA)
%INCLUDE(EVEDAUIB)
%INCLUDE(EVEDHLT1)
%INCLUDE(EVEDHLT2)
%INCLUDE(EVEDHLTA)
%INCLUDE(EVEDHLTB)
%INCLUDE(EVEDLMT1)
%INCLUDE(EVEDLMT2)
%INCLUDE(EVEDLMTA)
%INCLUDE(EVEDLMTB)
%INCLUDE(EVEDMSG1)
%INCLUDE(EVEDMSG2)
%INCLUDE(EVEDMSGA)
%INCLUDE(EVEDMSGB)
%INCLUDE(EVEDSTG1)
%INCLUDE(EVEDSTG2)
%INCLUDE(EVEDSTGA)
%INCLUDE(EVEDSTGB)
%INCLUDE(EVEDTIM1)
%INCLUDE(EVEDTIM2)
%INCLUDE(EVEDTIMA)
%INCLUDE(EVEDTIMB)
%INCLUDE(EVEDTRN1)
%INCLUDE(EVEDTRN2)
%INCLUDE(EVEDTRNA)
%INCLUDE(EVEDTRNB)
```

## EVEESA01—Short-on-Storage State/Action Table

```

/*****
/* (C) COPYRIGHT IBM CORP. 1990,1995 - ALL RIGHTS RESERVED */
/*
/* LICENSED MATERIALS - PROPERTY OF IBM */
/* RESTRICTED MATERIALS OF IBM */
/* 5685-151 */
/*
/* US GOVERNMENT USERS RESTRICTED RIGHTS -
/* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY
/* GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
/*
/*****
/*****
/* APAR# DATE */
/* -----
/* $01=PN52486 03/30/94 MP CICSAP SUPPORT FOR CICS V4
/*
/* PN36057 02/27/93 JS CICSAP TRAP FOR SHORT-ON-STORAGE FAILS IN
/* CICS V3.2 AS MSGDFHSM0100 AND MSGDFHSM0101
/* DO NOT HAVE 'I' SUFFIX
/*
/* PN17565 07/07/92 MP CICSAP SUPPORT FOR CICS V3R3
/*
/*****
/*****
/* CICS STATE ACTION TABLE FOR SHORT-ON-STORAGE RECOVERY */
/*
/* EVENTS:
/*
/* DFH0506 CICS IS UNDER STRESS, SHORT-ON-STORAGE
/* DFH0507 CICS IS NO LONGER SHORT-ON-STORAGE
/* DFHSM0100 applid CICS IS UNDER STRESS (SOS IN DSA)
/* DFHSM0101 applid CICS IS UNDER STRESS (SOS IN EDSA)
/* DFHSM0106 applid CICS IS NO LONGER SHORT-ON-STORAGE IN DSA
/* DFHSM0107 applid CICS IS NO LONGER SHORT-ON-STORAGE IN EDSA
/* DFHSM0116 applid CICS IS UNDER STRESS (SOS IN dsaname)
/* DFHSM0117 applid CICS IS NO LONGER SHORT ON STORAGE IN dsaname
/* DFHSM0131 applid CICS IS UNDER STRESS (SOS BELOW 16MB)
/* DFHSM0132 applid CICS IS NO LONGER SHORT ON STORAGE BELOW 16MB
/* DFHSM0133 applid CICS IS UNDER STRESS (SOS ABOVE 16MB)
/* DFHSM0134 applid CICS IS NO LONGER SHORT ON STORAGE ABOVE 16MB
/* DFHKE1800 CICS ABEND MESSAGE
/* RESET_MSG EXTERNAL RESET MESSAGE
/*
/* ACTION CLISTS:
/*
/* EVEES101 INCREMENT HALTED COUNTER
/* SET SUBSYSTEM STATUS TO HALTED IF UP
/* SCHEDULE SHORT-ON-STORAGE TIMER TO START EVEE104
/* EVEES102 DECREMENT HALTED COUNTER
/* SET SUBSYSTEM STATUS TO UP IF LAST HALTED CONDITION
/* CANCEL SHORT-ON-STORAGE TIMER
/* EVEES103 CANCEL SHORT-ON-STORAGE TIMER
/* EVEES104 (TRIGGERED BY EXPIRATION OF SHORT-ON-STORAGE TIMER)
/* ALERT OPERATOR
/* TAKE AN MVS DUMP OF TARGET CICS
/*****
PRODUCT=CICS
AREA=SOS
STATE=0
/*****

```



```

/* STATES: NORMAL ONE MSG TWO MSGS */
/* THREE MSGS FOUR MSGS FIVE MSGS */
/* STATE VALUES: 0 1 2 */
/*****/
EVENT=DFH0506 EVEES101/1 NA NA ,
 NA NA NA
EVENT=DFH0507 NOP EVEES102/0 NA ,
 NA NA NA
EVENT=DFHSM0100 EVEES101/1 /2 NA ,
 NA NA NA
EVENT=DFHSM0101 EVEES101/1 /2 NA ,
 NA NA NA
EVENT=DFHSM0106 NOP EVEES102/0 /1 ,
 NA NA NA
EVENT=DFHSM0107 NOP EVEES102/0 /1 ,
 NA NA NA
EVENT=DFHSM0116 EVEES101/1 /2 /3 ,
 /4 /5 NA
EVENT=DFHSM0117 NOP EVEES102/0 /1 ,
 /2 /3 /4
EVENT=DFHSM0131 EVEES101/1 /2 NA ,
 NA NA NA
EVENT=DFHSM0132 NOP EVEES102/0 /1 ,
 NA NA NA
EVENT=DFHSM0133 EVEES101/1 /2 NA ,
 NA NA NA
EVENT=DFHSM0134 NOP EVEES102/0 /1 ,
 NA NA NA
EVENT=CICSDN NOP EVEES103/0 EVEES103/0 ,
 EVEES103/0 EVEES103/0 EVEES103/0
EVENT=CICSINIT NOP EVEES103/0 EVEES103/0 ,
 EVEES103/0 EVEES103/0 EVEES103/0

```

## EVEESA02—Storage Violation State/Action Table

```

/*****
/* (C) COPYRIGHT IBM CORP. 1990, 1995 - ALL RIGHTS RESERVED */
/*
/* LICENSED MATERIALS - PROPERTY OF IBM */
/* RESTRICTED MATERIALS OF IBM */
/* 5685-151 */
/*
/* US GOVERNMENT USERS RESTRICTED RIGHTS -
/* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY
/* GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
/*
/*
/*
/*****
/*****
/* APAR# DATE */
/* -----
/*
/*****
/*****
/* CICS STATE ACTION TABLE FOR STORAGE VIOLATION PROCESSING */
/*
/* EVENTS:
/*
/* DFH0508 A STORAGE VIOLATION HAS OCCURRED */
/* DFHSM0102 applid A STORAGE VIOLATION (CODE X'code') HAS BEEN */
/* DETECTED IN MODULE modname */
/* START_TRACE SEVERITY OF ERROR DICTATES A TRACE BE STARTED */
/* RESET_MSG OPERATOR MESSAGE TO RETURN TO NORMAL STATE */
/*
/* ACTION CLISTS:
/*
/* EVEES201 USE THRESHOLDING TO DETERMINE SEVERITY OF ERROR */
/* GENALERT TO THE OPERATOR */
/* SEND "START TRACE" MESSAGE IF SEVERITY WARRANTS */
/* EVEES202 GENALERT TO THE OPERATOR */
/* TURN ON FAQE TO CAPTURE A BETTER DUMP */
/* TURN ON CICS TRACE */
/* PRINT OR COPY THE DUMP FOR PD ANALYSIS */
/* EVEES203 TURN OFF CICS TRACE */
/*****
PRODUCT=CICS
AREA=VIOL
STATE=0
/*****
/* STATES: NORMAL TRACE TRACE */
/* RUNNING RUN */
/* STATE VALUES: 0 1 2 */
/*****
EVENT=DFH0508 EVEES201/0 EVEES203/0
EVENT=DFHSM0102 EVEES201/0 EVEES203/0
EVENT=START_TRACE EVEES202/1 NA
EVENT=CICSINIT NA /0
EVENT=CICSDN NA /0

```

## EVEESA03—Autoinstall Problem Recovery State/Action Table

```

/*****
/* (C) COPYRIGHT IBM CORP. 1990,1995 - ALL RIGHTS RESERVED */
/* */
/* LICENSED MATERIALS - PROPERTY OF IBM */
/* RESTRICTED MATERIALS OF IBM */
/* 5685-151 */
/* */
/* US GOVERNMENT USERS RESTRICTED RIGHTS - */
/* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY */
/* GSA ADP SCHEDULE CONTRACT WITH IBM CORP. */
/* */
/* */
/*****
/*****
/* APAR# DATE */
/* ----- */
/* $01=PN58923 06/22/94 SW AFTER PTF UN63573, RECEIVES MSGEVE653E */
/* FAILED TO LOCATE AREA FOR MSGDFHZC5966, */
/* STATE ACTION TABLE NEEDS TO BE REVISED */
/* */
/* IW00083 06/07/91 Missing XA version of DFH5966I */
/*****
/*****
/* CICS STATE ACTION TABLE FOR AUTOINSTALL PROBLEM RECOVERY */
/* */
/* EVENTS: */
/* */
/* DFH3482 tranid time LOGON FROM NODE nodeid REJECTED, */
/* INSUFFICIENT STORAGE FOR AUTOINSTALL */
/* DFHZC3482 E date time applid tranid LOGON FROM NODE node @01A*/
/* REJECTED INSUFFICIENT STORAGE FOR AUTOINSTALL REQUEST@01a*/
/* sense ((instance) MODULE name: DFHZLGX | DFHZSCX) @01A*/
/* DFH3483 tranid time LOGON FROM NODE nodeid REJECTED, */
/* AUTOINSTALL LIMIT REACHED */
/* DFH5935I AUTOINSTALL FOR TERMINAL: termid, NETNAME netname, */
/* MODEL-NAME modelname, SUCCESSFUL */
/* DFHZC5966 date time INSTALL/DELETE/RESTORE STARTED FOR */
/* resource-type(termid) (MODULE NAME: modname) */
/* */
/* ACTION CLISTS: */
/* */
/* EVEES301 ALERT THE OPERATOR OF PROBLEM */
/* EVEES302 SCHEDULE AUTOINSTALL TIMER */
/* EVEES303 CANCEL AUTOINSTALL TIMER */
/* EVEES304 (TRIGGERED BY EXPIRATION OF AUTOINSTALL TIMER) */
/* ALERT THE OPERATOR OF PROBLEM */
/*****
PRODUCT=CICS
AREA=AUTO
STATE=0
/*****
/* STATES: NORMAL AUTOINSTALL */
/* FAILURE */
/* STATE VALUES: 0 1 */
/*****
EVENT=DFH3482 EVEES301/0 NOP
EVENT=DFHZC3482 EVEES301/0 NOP
EVENT=DFH3483 EVEES302/1 NOP
EVENT=DFH5935I NOP EVEES303/0
EVENT=DFH5966I NOP EVEES303/0
EVENT=DFHZC5966 NOP EVEES303/0
EVENT=CICSINIT NOP EVEES303/0
EVENT=CICSDN NOP EVEES303/0

```

## EVEESA04—VTAM ACB Problem Recovery State/Action Table

```

/*****
/* (C) COPYRIGHT IBM CORP. 1990,1995 - ALL RIGHTS RESERVED */
/*
/* LICENSED MATERIALS - PROPERTY OF IBM */
/* RESTRICTED MATERIALS OF IBM */
/* 5685-151 */
/*
/* US GOVERNMENT USERS RESTRICTED RIGHTS - */
/* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY */
/* GSA ADP SCHEDULE CONTRACT WITH IBM CORP. */
/*
/*****
/*****
/* APAR# DATE */
/* -----
/* PN34103 02/11/93 MP CICS AO DOES NOT KNOW VTAM ACB IS OPEN IF */
/* ROUTINE EVEEI002 NOT RUN AND CEMT SET VTAM */
/* OPEN ISSUED WHEN ACB ALREADY OPEN */
/*
/* PN07520 07/15/92 MP CICS AO WILL LOOP TRYING TO OPEN VTAM ACB IF*/
/* MISMATCH BETWEEN APPLID IN CONTROL FILE AND*/
/* THE ONE USED BY CICS */
/*
/* IW00028 05/30/91 Add MAJNUP, APPLUP from IST093I */
/*****
/*****
/* CICS STATE ACTION TABLE FOR CICS VTAM ACB PROBLEM RECOVERY */
/*****
/*
/* DERIVATION OF EVENTS FROM SYSTEM MESSAGES: */
/*
/* .DFH1500 xxx CONTROL IS BEING GIVEN TO CICS (for xa) */
/* .DFH1517 xxxxxxxx CONTROL IS BEING GIVEN TO CICS. (for esa) */
/* CICSUP <-- ALL */
/* .DFH1572 UNABLE TO OPEN VTAM ACB,rc=xxxxxxx,ACB CODE=yy */
/* VTAMDN <-- =52.,=5C. */
/* MAJNER <-- =5A. */
/* VTAMER <-- code = all others */
/* .DFHKE1799 */
/* .DFH1799 TERMINATION OF CICS IS COMPLETE */
/* EVEEI004 <-- */
/* .DFH2316I VTAM ACB IS CLOSED */
/* ACBCLS <-- all */
/* DFHCZ3463 or */
/* DFH3463I VTAM ACB opened. VTAM Return Code = X'zz'. */
/* TIME=hh:mm:ss.(MODULE NAME:modname) */
/* VTAMDN <-- X'52', X'5C',=52.,=5C. */
/* MAJNER <-- X'5A',=5A. */
/* ACBOPN <-- X'00',=00. */
/* VTAMER <-- code = all others */
/* .IST020I VTAM INITIALIZATION COMPLETE */
/* EVEES408 <-- all */
/* .IEF403I CICS INITIALIZATION */
/* EVEEI002 <-- all */
/* .EVE172I PPI open */
/* PPIOPN <-- all */
/* .IST102I VTAM DOWN */
/* EVEES409 <-- all */
/* .EVE780 MAJOR NODE UP */
/* EVEES411 <-- */
/* .EVE780 APPLID UP */
/* EVEES412 <--

```

```

/* */
/* ACTION CLISTS/EVENTS CREATED: */
/* */
/* EVEERINT resynchronization */
/* VTAMUP <-- If VTAM is up */
/* VTAMDN <-- if VTAM is not up */
/* ACBOPN <-- If acb is already open and connected */
/* EVEEI002 process IEF403I */
/* CICSINIT <-- */
/* EVEEI004 process '1799 */
/* CICSNDN <-- */
/* EVEES401 START VTAM MAJOR NODE */
/* EVEES409 <-- if VTAM is down */
/* *(one) MAJNUP <-- if previously up */
/* *(mult)MAJNUP <-- if successful in opening major node */
/* MAJNER <-- if unsuccessful in opening major node */
/* EVEES402 START VTAM APPLID */
/* EVEES409 <-- if VTAM is down */
/* APPLUP <-- if successful in opening APPLID */
/* APPLER <-- if unsuccessful in opening APPLID */
/* ACBOPN <-- */
/* EVEES403 CHECK IF CICS IS UP */
/* CICSUP <-- if CICS was found to be active */
/* EVEES404 SET GLOBAL TO ACB OPENED */
/* EVEES405 SET GLOBAL TO ACB CLOSED */
/* EVEES406 CHECK IF PPI TO CICS IS OPEN */
/* PPIOPN <-- if PPI was found to be open */
/* EVEES407 SEND ALERT MESSAGES FOR VTAMACB */
/* EVEES408 BROADCAST VTAM UP TO ALL CICS SUBSYSTEMS IN DOMAIN */
/* *(mult)VTAMUP <-- for all CICS subsystem in domain */
/* EVEES409 BROADCAST VTAM DOWN TO ALL CICS SUBSYSTEMS IN DOMAIN */
/* *(mult)VTAMDN <-- for all CICS subsystem in domain */
/* EVEES410 Open CICS ACB */
/* DFH3463 <-- prompts this message */
/* EVEES411 BROADCAST VTAM MAJOR NODE IS UP */
/* *(mult)MAJNUP <-- */
/* EVEES412 Send APPLID up message(not implemented) */
/* APPLUP <-- for cics */
/* EVEES415 Check APPLID for mismatch */
/* MAJNER <-- for cics */
/* */
/* EVENTS: */
/* VTAMUP VTAM UP */
/* VTAMDN VTAM DOWN */
/* VTAMER VTAM ERROR */
/* MAJNUP MAJOR NODE UP */
/* MAJNER MAJOR NODE ERROR */
/* APPLUP APPLID UP */
/* APPLER APPLID ERROR */
/* CICSINIT CICS INITIALIZING */
/* CICSUP CICS IS UP */
/* CICSNDN CICS IS DOWN */
/* PPIOPN CICS PPI IS ACTIVE */
/* ACBOPN CICS ACB IS OPEN */
/* ACBCLS CICS ACB IS CLOSED */
/* */
/*****
PRODUCT=CICS
AREA=VTAMACB
STATE=0
/*****
/* STATE VALUES: 0-INITIAL 1-VTAM 2-MAJOR NODE 3-APPLID */
/* ACTIONS ACTIONS ACTIONS */
/* REQUIRED REQUIRED REQUIRED */

```

```

/* 4-CICS INIT 5-CICS 6-PPI CHECK 7-ACB */
/* ACTIONS IN PROGRESS PENDING */
/* REQUIRED OPEN/CLOS*/
/* 8-ACB OPEN 9-ACB CLOSED */
/*
/*****
/* XXXXXXXX/N XXXXXXXX/N XXXXXXXX/N XXXXXXXX/N */
/*****
EVENT=VTAMUP EVEES401/2 EVEES401/2 NOP EVEES401/2 ,
 NOP EVEES401/2 EVEES401/2 NA ,
 NA NA
EVENT=VTAMDN EVEES405/1 NOP EVEES405/1 EVEES405/1 ,
 EVEES405/1 EVEES405/1 EVEES405/1 EVEES405/1 ,
 EVEES405/1 /1
EVENT=VTAMER EVEES407/1 EVEES407/1 EVEES407/1 EVEES407/1 ,
 EVEES407/1 EVEES407/1 EVEES407/1 EVEES407/1 ,
 EVEES407/1 EVEES407/1
EVENT=MAJNUP NOP NOP EVEES402/3 NOP ,
 NOP NOP NOP NOP ,
 NOP EVEES402/3
EVENT=MAJNER EVEES401/2 EVEES407/1 EVEES407/2 EVEES407/2 ,
 EVEES401/2 EVEES401/2 EVEES407/2 EVEES415/2 ,
 EVEES407/2 EVEES401/2
EVENT=APPLUP EVEES403/5 NA NA EVEES403/5 ,
 NA NA NA NA ,
 NA EVEES403/5
EVENT=APPLER EVEES407/3 NA NA EVEES407/3 ,
 NA NA NA NA ,
 NA NA
EVENT=CICSINIT /4 NOP NOP NOP ,
 NOP /4 NA NA ,
 NA NA
EVENT=CICSUP EVEES404/8 NOP NOP NOP ,
 EVEES404/8 EVEES406/6 NOP NOP ,
 NOP NA
EVENT=CICSDN EVEES405/5 EVEES405/1 EVEES405/2 EVEES405/3 ,
 EVEES405/5 EVEES405/5 EVEES405/5 EVEES405/5 ,
 EVEES405/5 EVEES405/5
EVENT=PPIOPN NA NA NOP NOP ,
 NA EVEES410/7 EVEES410/7 NOP ,
 NOP NA
EVENT=ACBOPN EVEES404/8 EVEES404/8 EVEES404/8 EVEES404/8 ,
 EVEES404/8 EVEES404/8 EVEES404/8 EVEES404/8 ,
 NOP EVEES404/8
EVENT=ACBCLS EVEES405/9 EVEES405/9 EVEES405/9 EVEES405/9 ,
 /9 /9 EVEES405/9 EVEES405/9 ,
 EVEES405/9 NOP

```

---

## EVEPRFAO—CICS Automation Operator Profiles

```

* (C) COPYRIGHT IBM CORP. 1990,1995 - ALL RIGHTS RESERVED *
* * *
* LICENSED MATERIALS - PROPERTY OF IBM *
* RESTRICTED MATERIALS OF IBM *
* 5685-151 *
* *
* US GOVERNMENT USERS RESTRICTED RIGHTS - *
* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY *
* GSA ADP SCHEDULE CONTRACT WITH IBM CORP. *

* APAR# DATE *
* -----
* IW00033 05/28/91 Correct handling of CTLDOWN and service periods * 00071000

* CATEGORY : INITIALIZATION *
* MESSAGE ID : NONE *
* DESCRIPTION : PROFILE FOR THE AUTOMATED OPERATOR *
* VARIABLES : NONE *
* INPUT : PASSED PARAMETERS *
* OUTPUT : PASSED PARAMETERS *
* WHERE CALLED : OPERATOR LOGON *
* CALLED : NONE *

EVEPRFAO PROFILE IC=EVEEIAAI
AUTH CTL=GLOBAL
OPCLASS 1,2
END
```

---

## EVECHLTH—Health Check Sample Program (COBOL Version)

```

* EVECHLTH - HEALTH CHECK SAMPLE PROGRAM (COBOL VERSION)
*
* THIS PROGRAM ATTEMPTS TO READ A RECORD FROM A FILE NAMED
* 'FILEA'. IF THE PROGRAM IS SUCCESSFUL IN READING THE RECORD,
* A POSITIVE ACKNOWLEDGEMENT (ACK) IS SENT BACK. OTHERWISE, A
* NEGATIVE ACKNOWLEDGEMENT (NACK) IS SENT BACK.

* (C) COPYRIGHT IBM CORP. 1990, 1995 - ALL RIGHTS RESERVED
*
* LICENSED MATERIALS - PROPERTY OF IBM
* RESTRICTED MATERIALS OF IBM
* 5685-151
*
* US GOVERNMENT USERS RESTRICTED RIGHTS -
* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY
* GSA ADP SCHEDULE CONTRACT WITH IBM CORP.

* APAR# DATE
* -----
*

IDENTIFICATION DIVISION.
PROGRAM-ID. EVECHLTH.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
*
* DATA-AREAS FOR RIDFLD
77 RID PIC 9(6) VALUE IS ZERO.
*
* FILEA RECORD DESCRIPT'N
* DFH$CFIL ON CICS 2.1.2
01 FILEA. COPY DFH0CFIL.
*
LINKAGE SECTION.
*
* COMMAREA FOR RESPONSE
01 DFHCOMMAREA. COPY EVECAREA.
*
PROCEDURE DIVISION.
 EXEC CICS HANDLE ABEND LABEL(PGMABND) END-EXEC.

* ESTABLISH CONDITIONS.
*

 EXEC CICS HANDLE CONDITION
 NOTFND(NOTFD) END-EXEC.
 EXEC CICS HANDLE CONDITION
 NOTOPEN(NOTOPEN) END-EXEC.
 EXEC CICS HANDLE CONDITION
 DISABLED(DISAB) END-EXEC.
 EXEC CICS HANDLE CONDITION
 NOTAUTH(NOTAUTH) END-EXEC.
 EXEC CICS HANDLE CONDITION
 IOERR(IOERROR) END-EXEC.
 EXEC CICS HANDLE CONDITION
 ENDFILE(TOOHIGH) END-EXEC.

* ESTABLISH STARTING POINT
*

* SET UP INITIAL KEY
 MOVE '000000' TO RID.

 EXEC CICS STARTBR DATASET('FILEA')
```



```

 RIDFLD(RID) END-EXEC.

* READ FIRST RECORD *
* AND SEND REPLY ACCORDINGLY. *

* READ NEXT RECORD
 EXEC CICS READNEXT INTO(FILEA)
 DATASET('FILEA')
 RIDFLD(RID) END-EXEC.

* RID>NEXT FPAGE

* READ RECORD SUCCESSFULLY *

* SEND ACK
SEND-ACK.
 MOVE 'ACK' TO EVECAREA-RESP-TYPE.
 EXEC CICS RETURN END-EXEC.

* HANDLE END OF FILE CONDITIONS *

TOOHIGH.
* SEND NACK WITH NACK DATA OF
* TOOHIGH
 MOVE 'NACK' TO EVECAREA-RESP-TYPE.
 MOVE 'RECORD NUMBER TOO HIGH.'
 TO EVECAREA-RESP-DATA.
 EXEC CICS RETURN END-EXEC.

* HANDLE GENERAL CONDITIONS *

NOTFD.
* SEND NACK WITH NACK DATA OF
* ENDOFFILE
 MOVE 'NACK' TO EVECAREA-RESP-TYPE.
 MOVE 'CICS FILE NOT FOUND.'
 TO EVECAREA-RESP-DATA.
 EXEC CICS RETURN END-EXEC.
NOTOPEN.
* SEND NACK WITH NACK DATA OF
* FILE NOT OPENED
 MOVE 'NACK' TO EVECAREA-RESP-TYPE.
 MOVE 'CICS FILE NOT OPENED.'
 TO EVECAREA-RESP-DATA.
 EXEC CICS RETURN END-EXEC.
DISAB.
* SEND NACK WITH NACK DATA OF
* FILE DISABLED
 MOVE 'NACK' TO EVECAREA-RESP-TYPE.
 MOVE 'CICS FILE DISABLED.'
 TO EVECAREA-RESP-DATA.
 EXEC CICS RETURN END-EXEC.
NOTAUTH.
* SEND NACK WITH NACK DATA OF
* NOT AUTHORIZED TO BROWSE FILE
 MOVE 'NACK' TO EVECAREA-RESP-TYPE.
 MOVE 'NOT AUTHORIZED TO BROWSE CICS FILE.'
 TO EVECAREA-RESP-DATA.
 EXEC CICS RETURN END-EXEC.
IOERROR.
* SEND NACK WITH NACK DATA OF
* I/O ERROR
 MOVE 'NACK' TO EVECAREA-RESP-TYPE.
 MOVE 'I/O ERROR DETECTED BY CICS TRANSACTION.'
 TO EVECAREA-RESP-DATA.

```

```
EXEC CICS RETURN END-EXEC.
PGMABND.
* SEND NACK WITH NACK DATA OF
* PROGRAM ABENDED
MOVE 'NACK' TO EVECAREA-RESP-TYPE.
MOVE 'TRANSACTION ABENDED DURING EXECUTION...'
 TO EVECAREA-RESP-DATA.
EXEC CICS RETURN END-EXEC.
GOBACK.
```

---

## EVE CAREA—Layout of the Health Checking DFHCOMMAREA

```

* (C) COPYRIGHT IBM CORP. 1990,1995 - ALL RIGHTS RESERVED
*
* LICENSED MATERIALS - PROPERTY OF IBM
* RESTRICTED MATERIALS OF IBM
* 5685-151
*
* US GOVERNMENT USERS RESTRICTED RIGHTS -
* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY
* GSA ADP SCHEDULE CONTRACT WITH IBM CORP.

* APAR# DATE
* -----
*

*
* LAYOUT OF THE HEALTH CHECK RESPONSE AREA
*
* . RESP-TYPE IS THE COMPLETION CODE, EITHER -
* . 'ACK' - FOR A SUCCESSFUL HEALTH CHECK
* . 'NACK' - FOR AN UNSUCCESSFUL HEALTH CHECK
* . WHICH IS FILLED IN BY THE USER-WRITTEN
* . HEALTH CHECKING PROGRAM
* . RESP-DATA IS UP TO 100 BYTES OF MESSAGE DATA SENT BACK WITH
* . NACK RESPONSE
02 EVE CAREA-RESP-TYPE PIC X(4).
02 EVE CAREA-RESP-DATA PIC X(100).
```

## EVENTASK—PPI Initialization Member (NetView)

```
***** 00010000
* (C) COPYRIGHT IBM CORP. 1990,1994,1998 - ALL RIGHTS RESERVED * 00020000
* * 00020000
* LICENSED MATERIALS - PROPERTY OF IBM * 00020000
* RESTRICTED MATERIALS OF IBM * 00020000
* 5685-151 * 00020000
* * 00020000
* US GOVERNMENT USERS RESTRICTED RIGHTS - * 00020000
* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY * 00020000
* GSA ADP SCHEDULE CONTRACT WITH IBM CORP. * 00020000
***** 00010000
* * 00020000
* NAME: * 00030000
* * 00040000
* EVENTASK * 00050000
* * 00060000
* * 00070000
* TITLE: * 00080000
* * 00090000
* EVENTASK INITIALIZATION MEMBER * 00100000
* * 00110000
* * 00120000
* COMPONENT: * 00130000
* * 00140000
* CICS AUTOMATION OPTION * 00150000
* * 00160000
* * 00170000
* ENVIRONMENT: * 00180000
* * 00190000
* CAO RELEASE 1.0 * 00200000
* * 00210000
* * 00220000
* DESCRIPTION: * 00230000
* * 00240000
* THIS SAMPLE CONTAINS SOME DEFINITIONS FOR THE * 00250000
* EVENTASK INITIALIZATION MEMBERS. THE MEMBER * 00260000
* MUST BE CREATED IN (ONE OF) THE DSIPARM DATA * 00270000
* SETS. * 00280000
* * 00290000
* * 00300000
* NOTES: * 00310000
* DEPENDENCIES: * 00320000
* NONE. * 00330000
* OTHER: * 00340000
* THIS SAMPLE MUST BE CUSTOMIZED ACCORDING TO THE * 00350000
* ENVIRONMENT OF THE INSTALLATION. * 00360000
* * 00370000
* * 00380000
* AUTHOR: LDR * 00390000
* * 00400000
* DATE: 01/02/91 * 00440000
* * 00450000
* * 00470000
* * 00480000
***** 00490000
* 00500000
* 00510000
***** 00520000

* APAR# DATE *
* ----- *
* IW00019 06/05/91 ADD NACK RESPONSE FOR REMOTE DOMAIN RESPONSE *
```

```

* IW00088 06/11/91 ADD HEALTH SERVER RESPONSE *
* $01=0W35605,V1R4,22OCT98,APC(JK): CICS WAVE 2 *

***** 00610000
* 00620000
* 00630000
* AN ASTERISK IN COLUMN ONE INDICATES A COMMENT LINE. 00640000
* 00650000
* 00660000
***** 00670000
* 00680000
* PPI BUFFER QUEUE LIMIT FOR EVENTASK * 00690000
* 00700000
* SPECIFY A 2-DIGIT OR 3-DIGIT NUMERIC VALUE. THE * 00710000
* MINIMUM VALUE IS 10, THE MAXIMUM VALUE IS 999. * 00720000
* IF OMITTED, A VALUE OF 15 IS ASSUMED. * 00730000
* 00740000
***** 00750000
* 00760000
BUFFQL=20 00770000
* 00780000
***** 00790000
* 00800000
* PPI RECEIVER IDENTIFIER FOR EVESNPPI TASK * 00810000
* 00820000
* IF OMITTED, NETVPPI IS ASSUMED. * 00830000
* 00840000
***** 00850000
* 00860000
RECEIVERID=NETVCPPI 00870000
* 00880000
***** 00890000
* 00900000
* FUNCTION NAMES AND SERVER PROGRAMS * 00910000
* 00920000
* THE FUNCTION NAME MAY NOT START WITH "EVE". * 00930000
* AT LEAST ONE VALID SERVER MUST BE SPECIFIED. * 00930000
* 00940000
***** 00950000
* 00960000
SERVER=REQUEST,LMT,AUTCPPI,EVEEYPPS 00970000
SERVER=RESPONSE,CEMT,AUTCPPI,EVESNRSP 00970000
SERVER=RESPONSE,LMT,AUTCPPI,EVESNRSP 00970000
* @01A
SERVER=RESPONSE,RLS,AUTCPPI,EVESNRSP 00970000
* 01000000
SERVER=REQUEST,NACK,AUTCPPI,EVESNACK 00970000
SERVER=RESPONSE,NACK,AUTCPPI,EVESNACK 00970000
* 01000000
***** 01010000
* 01020000
* END OF EVENTASK INITIALIZATION MEMBER * 01030000
* 01040000

```

## EVESPINM—PPI Initialization Member (CICS)

```

EVESPINM TITLE '*** CAO *** CPDS INITIALIZATION TABLE - PROLOGUE *00010000
 **' 00020000
EVESPINM CSECT , ESTABLISH CONTROL SECTION 00030000
SPACE 1 00040000
***** 00050000
* * 00060000
* * 00070000
* NAME: * 00080000
* EVESPINM * 00090000
* * 00100000
* * 00110000
* TITLE: * 00120000
* CPDS INITIALIZATION TABLE * 00130000
* * 00140000
* * 00150000
* COMPONENT: * 00160000
* * 00170000
* CICS AUTOMATION OPTION (CAO) * 00180000
* * 00190000
* COPYRIGHT= * 00200000
* LICENSED MATERIALS - PROPERTY OF IBM * 00210000
* RESTRICTED MATERIALS OF IBM * 00220000
* 5685-151 * 00230000
* (C) COPYRIGHT IBM CORP. 1990, 1995, 1998 * @01C 00240001
* * 00250000
* US GOVERNMENT USERS RESTRICTED RIGHTS - * 00260000
* USE, DUPLICATION OR DISCLOSURE RESTRICTED BY * 00270000
* GSA ADP SCHEDULE CONTRACT WITH IBM CORP. * 00280000
* * 00290000
* * 00300000
* ENVIRONMENT: * 00310000
* * 00320000
* CAO RELEASE 1.0 * 00330000
* * 00340000
* * 00350000
* FUNCTION: * 00360000
* * 00370000
* TO GENERATE THE CPDS INITIALIZATION TABLE. THIS * 00380000
* TABLE DEFINES AMONG OTHERS THE RELATIONSHIP * 00390000
* BETWEEN FUNCTION NAMES AND CICS TRANSACTION * 00400000
* NAMES. * 00410000
* * 00420000
* * 00430000
* * 00440000
* ENTRY POINTS: * 00450000
* * 00460000
* EVELPINM * 00470000
* * 00480000
* * 00490000
* INVOCATION: * 00500000
* * 00510000
* N/A * 00520000
* * 00530000
* * 00540000
* INPUT: * 00550000
* * 00560000
* N/A * 00570000
* EJECT * 00580000
* * 00590000
* * 00600000
* PROCESS: * 00610000
* * 00620000

```

|   |                                                   |          |          |
|---|---------------------------------------------------|----------|----------|
| * | N/A                                               | *        | 00630000 |
| * |                                                   | *        | 00640000 |
| * |                                                   | *        | 00650000 |
| * | OUTPUT:                                           | *        | 00660000 |
| * |                                                   | *        | 00670000 |
| * | NONE, THERE IS NO EXECUTABLE CODE.                | *        | 00680000 |
| * |                                                   | *        | 00690000 |
| * |                                                   | *        | 00700000 |
| * | ERRORS DETECTED:                                  | *        | 00710000 |
| * |                                                   | *        | 00720000 |
| * | THE FOLLOWING MNOTES FROM EVEMPINM MAY BE ISSUED: | *        | 00730000 |
| * |                                                   | *        | 00740000 |
| * | 8,'INCORRECT TYPE SPECIFICATION'                  | *        | 00750000 |
| * | 8,'SEQUENCE ERROR IN EVEMPINM SPECIFICATIONS'     | *        | 00760000 |
| * | 8,'REQUIRED PARAMETER(S) OMITTED FOR TYPE=ENTRY'  | *        | 00770000 |
| * | 8,'INCORRECT FUNCTION OR TRANSID SPECIFICATION'   | *        | 00780000 |
| * | 4,'INCORRECT BUFFQL SPECIFICATION, 3 ASSUMED'     | *        | 00790000 |
| * | 4,'INCORRECT RECEIVERID SPECIFICATION, NETVCPPI   | *        | 00800000 |
| * | ASSUMED'                                          | *        | 00810000 |
| * | 4,'INCORRECT CONSOLE SPECIFICATION, SPECIFICATION | * PFL011 | 00820000 |
| * | IGNORED'                                          | * PFL011 | 00830000 |
| * | 4,'DUPLICATE FUNCTION SPECIFICATION IGNORED'      | *        | 00840000 |
| * | 0,'BUFFQL, RECEIVERID, FUNCTION, TRANSID, AND     | * PFL011 | 00850000 |
| * | CONSOLE ARE IGNORED FOR TYPE=DSECT'               | * PFL011 | 00860000 |
| * | 0,'FUNCTION AND TRANSID ARE IGNORED FOR           | *        | 00870000 |
| * | TYPE=INITIAL'                                     | *        | 00880000 |
| * | 0,'BUFFQL, RECEIVERID, AND CONSOLE ARE IGNORED    | * PFL011 | 00890000 |
| * | FOR TYPE=ENTRY'                                   | * PFL011 | 00900000 |
| * | 0,'BUFFQL, RECEIVERID, FUNCTION, TRANSID, AND     | * PFL011 | 00910000 |
| * | CONSOLE ARE IGNORED FOR TYPE=FINAL'               | * PFL011 | 00920000 |
| * |                                                   | *        | 00930000 |
| * |                                                   | *        | 00940000 |
| * | EXITS                                             | *        | 00950000 |
| * |                                                   | *        | 00960000 |
| * | N/A                                               | *        | 00970000 |
| * |                                                   | *        | 00980000 |
| * |                                                   | *        | 00990000 |
| * | REGISTER USAGE:                                   | *        | 01000000 |
| * |                                                   | *        | 01010000 |
| * | N/A                                               | *        | 01020000 |
| * | EJECT                                             | *        | 01030000 |
| * |                                                   | *        | 01040000 |
| * |                                                   | *        | 01050000 |
| * | NOTES                                             | *        | 01060000 |
| * |                                                   | *        | 01070000 |
| * | ATTRIBUTES:                                       | *        | 01080000 |
| * | RENT, AMODE(31),RMODE(ANY), AC=0.                 | *        | 01090000 |
| * | RESTRICTIONS:                                     | *        | 01100000 |
| * | DEPENDENCIES:                                     | *        | 01110000 |
| * | OTHER:                                            | *        | 01120000 |
| * | EVEPINM IS TO BE LINK-EDITED IN ONE OF THE        | *        | 01130000 |
| * | CICS DFHRPL LIBRARIES.                            | *        | 01140000 |
| * |                                                   | *        | 01150000 |
| * | GENERATION OF THE CPDS INITIALIZATION TABLE       | *        | 01160000 |
| * | REQUIRES EVEMPINM TYPE=DSECT TO BE SPECIFIED.     | *        | 01170000 |
| * |                                                   | *        | 01180000 |
| * |                                                   | *        | 01190000 |
| * | AUTHOR:                                           | *        | 01200000 |
| * |                                                   | *        | 01210000 |
| * | LDR                                               | *        | 01220000 |
| * |                                                   | *        | 01230000 |
| * |                                                   | *        | 01240000 |
| * | DATE:                                             | *        | 01250000 |
| * |                                                   | *        | 01260000 |

```

* 01/04/91
*
*

 EJECT
 SPACE 1

*
*
* MODIFICATIONS
* =====
*
* DATE DESCRIPTION PGMR
* =====
* 02/25/91 CONSOLE= SUPPORT. LEO DE REUS
*
*

* APAR# DATE
* -----
* $L1 07/18/95 CG R4 OF AOC/CICS
* IW000088 06/11/91 ENHANCE HEALTH CHECKING INTERFACE
* IW000091 06/11/91 CORRECT RECIEVERID COMMENTS
* $01=OW35605,V1R4,22OCT98,APC(JK): CICS WAVE 2

 TITLE '*** CAO *** CPDS INITIALIZATION TABLE - GENERATE TABLE
 ***'

*
* SPECIFY DEFINITIONS FOR CPDS INITIALIZATION TABLE
*

 SPACE 2

*
* INITIAL EVEMPINM SPECIFICATION
*
* 'INITIAL' MUST BE THE FIRST EVEMPINM TYPE SPECIFIED.
*
* BUFFQL SPECIFIES THE BUFFER QUEUE LIMIT FOR THE
* CICS 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 EVESNPPI INITIALIZATION MEMBER.
* IF THIS KEYWORD IS OMITTED, NETVCPPI IS ASSUMED.
*
* CONSOLE SPECIFIES THE 1 TO 4-CHARACTER TERMINAL
* IDENTIFIER OF THE CONSOLE ON WHICH THE LONG-RUNNING
* COPC TRANSACTION IS STARTED. IF THIS SPECIFICATION
* IS OMITTED, COPC IS STARTED WITHOUT A TERMINAL.
*

 SPACE 1
 EVEMPINM TYPE=INITIAL, INITIAL ENTRY
 BUFFQL=4, BUFFER QUEUE LIMIT
 RECEIVERID=NETVCPPI, NPDS RECEIVER IDENTIFICATION
 USERID=YES, USE USERID IN INVOCATION
 CONSOLE= TERMID OF CONSOLE FOR COPC
 EJECT

```



```

SPACE 1

*
* DEFINE FUNCTION AND TRANSACTION NAMES
*
* SPECIFY FOR A FUNCTION THE CICS TRANSACTION TO
* BE EXECUTED WHEN THE FUNCTION IS REQUESTED.
* THE FUNCTION NAME CAN BE FROM 1 TO 8 CHARACTERS,
* AND MUST NOT START WITH THE CHARACTERS "EVE".
*

SPACE 1
EVEPINM TYPE=ENTRY, DEFINE A FUNCTION
 FUNCTION=CEMT, FUNCTION NAME
 TRANSID=COMT TRANSACTION NAME
SPACE 2
EVEPINM TYPE=ENTRY, DEFINE A FUNCTION
 FUNCTION=LMT, FUNCTION NAME
 TRANSID=COLR TRANSACTION NAME
SPACE 2
EVEPINM TYPE=ENTRY, DEFINE A FUNCTION
 FUNCTION=HEALTH, FUNCTION NAME
 TRANSID=COHR TRANSACTION NAME
SPACE 2
EVEPINM TYPE=ENTRY, DEFINE A FUNCTION
 FUNCTION=RLS, FUNCTION NAME
 TRANSID=CORL TRANSACTION NAME
SPACE 2
EXAMPLE OF USER DEFINED ENTRY
*
* EVEPINM TYPE=ENTRY, DEFINE A FUNCTION
* FUNCTION=TEST, FUNCTION NAME
* TRANSID=TSTL TRANSACTION NAME
EJECT
SPACE 1

*
* 'FINAL' MUST BE THE LAST EVEPINM TYPE SPECIFIED.
*

SPACE 1
EVEPINM TYPE=FINAL REQUIRED END
TITLE '*** CAO *** CPDS INITIALIZATION TABLE - DUMMY SECTION
 ** '
SPACE 1
EVEPINM TYPE=DSECT
EJECT
SPACE 1
END EVELPINM

```

## EVEMEYU1—CPSM Messages (Included by EVEMSG01)

```

***** 00000100
* (C) COPYRIGHT IBM CORP. 1998 - ALL RIGHTS RESERVED 00000200
***** 00000300
* 00000400
* LICENSED MATERIALS - PROPERTY OF IBM 00000500
* 5685-151 (C) Copyright IBM Corp. 1998 00000600
* 00000700
***** 00000800
***** 00000900
* APAR# DATE * 00001000
* ----- * 00001100
* $L0=OW35605,V1R4,12OCT98,APC(JK): CICS Wave 2 Changes. * 00001200
***** 00001300
***** 00001400
* * 00001500
* Start of BBM message block * 00001600
* * 00001700
***** 00001800
* 00001900
IF MSGID='BBM' . 00002000
 THEN 00002100
 BEGIN; 00002200
 00002300
* CAS Startup complete 00002400
IF MSGID='BBMZA001I' & 00002500
 TOKEN(4) = 'INITIALIZATION' 00002600
THEN EXEC(CMD('ACTIVMSG UP=YES') ROUTE(ALL *)) 00002700
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00002800
 00002900
* CAS Shutdown complete 00002910
IF MSGID='BBMZA999I' 00002920
THEN EXEC(CMD('TERMMSG FINAL=YES') ROUTE(ALL *)) 00002940
 DISPLAY(N) BEEP(N) HOLD(N) NETLOG(Y) SYSLOG(Y); 00002950
 00002960
***** 00003000
* * 00003100
* Trap unknown replies * 00003200
* * 00003300
***** 00003400
 00003500
 IF IFRAUWF1(6) = '1' & DOMAINID = %AOFDOM% 00003600
 THEN EXEC(CMD('OUTREP ')ROUTE(ONE %AOFOPWATORS%)); 00003700
 00003800
***** 00003900
* END OF BBM Message Block * 00004000
***** 00004100
 ALWAYS; 00004200
 END; 00004300
***** 00004400
* * 00004500
* Start of EYU message block * 00004600
* * 00004700
***** 00004800
* 00004900
IF MSGID='EYU' . 00005000
 THEN 00005100
 BEGIN; 00005200
 00005300
***** 00005400
* * 00005500
* Trap unknown replies * 00005600

```

```

*

 IF IFRAUWF1(6) = '1' & DOMAINID = %AOFDOM%
 THEN EXEC(CMD('OUTREP ')ROUTE(ONE %AOFOPWTORS%));

* END OF CICS AUTOMATION EVEMEYU1 MESSAGE TABLE ENTRIES

 ALWAYS;
END;

```



## Appendix D. CICS Automation SMF Record Layout

SMF records are created by EVESUSMF. The record layout is as follows:

| NAME     | DIS | LENGTH | DESCRIPTION                                   |
|----------|-----|--------|-----------------------------------------------|
| SMFRLEN  | 000 | 002    | AL2(NNN) TOTAL RECORD LENGTH                  |
| SMFSEGD  | 002 | 002    | X'0000'                                       |
| SMFHFLAG | 004 | 001    | X'80'                                         |
|          |     |        | X'8E' FOR MVS/ESA                             |
| SMFRTYPE | 005 | 001    | SMF RECORD TYPE FROM SMFID(X'XY')             |
| SMFTIME  | 006 | 004    | TIME SINCE MIDNIGHT (HUNDREDS OF SECONDS)     |
| SMFDATE  | 010 | 004    | DATE IN FORMAT X'0CYYDDDF' OR X'00YYDDDF'     |
| SMFCASID | 014 | 004    | SYSTEM IDENTIFIER FROM SMCASID                |
| SMFSSID  | 018 | 004    | C'CAO '                                       |
| *        | 022 |        | END OF STANDARD SMF HEADER                    |
| SMFRVER  | 022 | 002    | X'0001' (RECORD VERSION)                      |
| SMFDOMID | 024 | 008    | DOMAIN IDENTIFIER FROM MVTCURAN               |
| SMFSUBSN | 032 | 008    | SUBSYS PADDED WITH BLANKS                     |
| SMFPROCN | 040 | 008    | PROCNM PADDED WITH BLANKS                     |
| SMFSAPPL | 048 | 008    | SPAPPL PADDED WITH BLANKS                     |
| SMFGAPPL | 056 | 008    | GNAPPL PADDED WITH BLANKS                     |
| SMFETYPE | 064 | 001    | EVENT TYPE FROM EVTYPE                        |
|          |     |        | X'01' CICS START-UP TIME (START)              |
|          |     |        | X'02' CICS SERVICE START-UP TIME (UP)         |
|          |     |        | X'03' CICS SERVICE END TIME (STOP)            |
|          |     |        | X'04' CICS END TIME (DOWN)                    |
|          |     |        | X'05' CICS SERVICE IN PROBLEMS (HALTED)       |
|          |     |        | X'06' CICS SERVICE OUT OF PROBLEMS (UNHALTED) |
|          | 065 | 007    | XL7'00' (RESERVED)                            |
| SMFEDATA | 072 | XXX    | EVDATA (0-100 BYTES)                          |
| *        | 072 |        | MINIMUM SMF RECORD LENGTH                     |
| *        | 172 |        | MAXIMUM SMF RECORD LENGTH                     |



## Appendix E. Worksheets

Use the worksheets in this section to collect information that you will use for CICS Automation customization. It is recommended that you copy these pages from the book and work with the copies.

### NetView Domains and CICS Regions

List all of the CICS regions in each NetView domain. "Subsystem name" is the name you will use to identify the CICS region to CICS Automation.

**MVS system** \_\_\_\_\_ **NetView domain** \_\_\_\_\_

| CICS regions |                |              |                |
|--------------|----------------|--------------|----------------|
| MVS job name | Subsystem name | MVS job name | Subsystem name |
| _____        | _____          | _____        | _____          |
| _____        | _____          | _____        | _____          |
| _____        | _____          | _____        | _____          |
| _____        | _____          | _____        | _____          |
| _____        | _____          | _____        | _____          |

**MVS system** \_\_\_\_\_ **NetView domain** \_\_\_\_\_

| CICS regions |                |              |                |
|--------------|----------------|--------------|----------------|
| MVS job name | Subsystem name | MVS job name | Subsystem name |
| _____        | _____          | _____        | _____          |
| _____        | _____          | _____        | _____          |
| _____        | _____          | _____        | _____          |
| _____        | _____          | _____        | _____          |
| _____        | _____          | _____        | _____          |

NetView Domains and CICS Regions

MVS system \_\_\_\_\_ NetView domain \_\_\_\_\_

| CICS regions |                |              |                |
|--------------|----------------|--------------|----------------|
| MVS job name | Subsystem name | MVS job name | Subsystem name |
| _____        | _____          | _____        | _____          |
| _____        | _____          | _____        | _____          |
| _____        | _____          | _____        | _____          |
| _____        | _____          | _____        | _____          |
| _____        | _____          | _____        | _____          |

MVS system \_\_\_\_\_ NetView domain \_\_\_\_\_

| CICS regions |                |              |                |
|--------------|----------------|--------------|----------------|
| MVS job name | Subsystem name | MVS job name | Subsystem name |
| _____        | _____          | _____        | _____          |
| _____        | _____          | _____        | _____          |
| _____        | _____          | _____        | _____          |
| _____        | _____          | _____        | _____          |
| _____        | _____          | _____        | _____          |

MVS system \_\_\_\_\_ NetView domain \_\_\_\_\_

| CICS regions |                |              |                |
|--------------|----------------|--------------|----------------|
| MVS job name | Subsystem name | MVS job name | Subsystem name |
| _____        | _____          | _____        | _____          |
| _____        | _____          | _____        | _____          |
| _____        | _____          | _____        | _____          |
| _____        | _____          | _____        | _____          |
| _____        | _____          | _____        | _____          |



## Basic CICS Subsystem Information Sheets

**For domain** \_\_\_\_\_

[illegible]

## Basic CICS Subsystem Information Sheets

**For domain**

[illegible]

## Basic CICS Subsystem Information Sheets

**For domain** \_\_\_\_\_

[illegible]

## Automation Operator Information Sheets

Use the following to define the automation operators used specifically for CICS automation.

**Automation operators for domain \_\_\_\_\_**

| AUTOOPS Keyword | Operator ID | CICS Subsystem | Used for                                                                                                                                                                                                                                                                                                                                  |
|-----------------|-------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CICSMSTR=       |             | N/A            | This keyword is used to identify the automation operator that provides message distribution services for CICS subsystems. The message content is used to identify the subsystem for which this message applies. Messages are then routed to the automation operators defined with the CICSOP $nn$ keywords, which are subsystem specific. |
| CICSCPPI=       |             | N/A            | Handles program-to-program interface communications from CICS. Refer to "CICS Automation's Use of the Program-to-Program Interface" on page 6 for more information.                                                                                                                                                                       |
| CICSOP01=       | _____       | _____          | These keywords are used to identify the automation operators that handle messages for specific CICS subsystems. Refer to the AUTOOPS keyword in the control file entry described in "CICSCNTL - CICS extensions for SUBSYSTEM entry" on page 66 to see how the CICSOP $nn$ keyword is associated with a specific CICS subsystem.          |
| CICSOP02=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP03=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP04=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP05=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP06=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP07=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP08=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP09=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP10=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP11=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP12=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP13=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP14=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP15=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP16=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP17=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |

**Automation operators for domain \_\_\_\_\_**

| <b>AUTOOPS Keyword</b> | <b>Operator ID</b> | <b>CICS Subsystem</b> | <b>Used for</b>                                                                                                                                                                                                                                                                                                                           |
|------------------------|--------------------|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CICSMSTR=              |                    | N/A                   | This keyword is used to identify the automation operator that provides message distribution services for CICS subsystems. The message content is used to identify the subsystem for which this message applies. Messages are then routed to the automation operators defined with the CICSOP $nn$ keywords, which are subsystem specific. |
| CICSCPPI=              |                    | N/A                   | Handles program-to-program interface communications from CICS. Refer to "CICS Automation's Use of the Program-to-Program Interface" on page 6 for more information.                                                                                                                                                                       |
| CICSOP01=              | _____              | _____                 | These keywords are used to identify the automation operators that handle messages for specific CICS subsystems. Refer to the AUTOOPS keyword in the control file entry described in "CICSCNTL - CICS extensions for SUBSYSTEM entry" on page 66 to see how the CICSOP $nn$ keyword is associated with a specific CICS subsystem.          |
| CICSOP02=              | _____              | _____                 |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP03=              | _____              | _____                 |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP04=              | _____              | _____                 |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP05=              | _____              | _____                 |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP06=              | _____              | _____                 |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP07=              | _____              | _____                 |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP08=              | _____              | _____                 |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP09=              | _____              | _____                 |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP10=              | _____              | _____                 |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP11=              | _____              | _____                 |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP12=              | _____              | _____                 |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP13=              | _____              | _____                 |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP14=              | _____              | _____                 |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP15=              | _____              | _____                 |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP16=              | _____              | _____                 |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP17=              | _____              | _____                 |                                                                                                                                                                                                                                                                                                                                           |

## Automation Operator Information Sheets

### Automation operators for domain \_\_\_\_\_

| AUTOOPS Keyword | Operator ID | CICS Subsystem | Used for                                                                                                                                                                                                                                                                                                                                  |
|-----------------|-------------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CICSMSTR=       |             | N/A            | This keyword is used to identify the automation operator that provides message distribution services for CICS subsystems. The message content is used to identify the subsystem for which this message applies. Messages are then routed to the automation operators defined with the CICSOP $nn$ keywords, which are subsystem specific. |
| CICSCPPI=       |             | N/A            | Handles program-to-program interface communications from CICS. Refer to "CICS Automation's Use of the Program-to-Program Interface" on page 6 for more information.                                                                                                                                                                       |
| CICSOP01=       | _____       | _____          | These keywords are used to identify the automation operators that handle messages for specific CICS subsystems. Refer to the AUTOOPS keyword in the control file entry described in "CICSCNTL - CICS extensions for SUBSYSTEM entry" on page 66 to see how the CICSOP $nn$ keyword is associated with a specific CICS subsystem.          |
| CICSOP02=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP03=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP04=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP05=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP06=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP07=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP08=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP09=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP10=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP11=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP12=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP13=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP14=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP15=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP16=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |
| CICSOP17=       | _____       | _____          |                                                                                                                                                                                                                                                                                                                                           |

CICS Subsystem Group Information Sheet

CICS subsystems can be grouped. These groups can cross NetView domains. The purpose of this is to allow the operator to start or shut down a group of subsystems.

**Note:** You must have single-point-of-control defined to be able to start up or shut down subsystems across domains.

| Group name | CICS subsystems for this group |  |  |  |  |
|------------|--------------------------------|--|--|--|--|
|            |                                |  |  |  |  |
|            |                                |  |  |  |  |
|            |                                |  |  |  |  |
|            |                                |  |  |  |  |
|            |                                |  |  |  |  |
|            |                                |  |  |  |  |
|            |                                |  |  |  |  |
|            |                                |  |  |  |  |
|            |                                |  |  |  |  |
|            |                                |  |  |  |  |
|            |                                |  |  |  |  |
|            |                                |  |  |  |  |
|            |                                |  |  |  |  |
|            |                                |  |  |  |  |
|            |                                |  |  |  |  |
|            |                                |  |  |  |  |

## CICS Subsystem Group Information Sheet

[illegible]



## Service Periods Information Sheets

Write down the times (from-to) that this subsystem should be in service. Up to five time slots can be defined for each day. (Read "SERVICE - Service period definitions" on page 83 before filling out this information.)

**CICS subsystem name:** \_\_\_\_\_

| Day       | From-to | From-to | From-to | From-to | From-to |
|-----------|---------|---------|---------|---------|---------|
| Daily     |         |         |         |         |         |
| Weekday   |         |         |         |         |         |
| Weekend   |         |         |         |         |         |
| Monday    |         |         |         |         |         |
| Tuesday   |         |         |         |         |         |
| Wednesday |         |         |         |         |         |
| Thursday  |         |         |         |         |         |
| Friday    |         |         |         |         |         |
| Saturday  |         |         |         |         |         |
| Sunday    |         |         |         |         |         |

**CICS subsystem name:** \_\_\_\_\_

| Day       | From-to | From-to | From-to | From-to | From-to |
|-----------|---------|---------|---------|---------|---------|
| Daily     |         |         |         |         |         |
| Weekday   |         |         |         |         |         |
| Weekend   |         |         |         |         |         |
| Monday    |         |         |         |         |         |
| Tuesday   |         |         |         |         |         |
| Wednesday |         |         |         |         |         |
| Thursday  |         |         |         |         |         |
| Friday    |         |         |         |         |         |
| Saturday  |         |         |         |         |         |
| Sunday    |         |         |         |         |         |

**CICS subsystem name:** \_\_\_\_\_

| Day       | From-to | From-to | From-to | From-to | From-to |
|-----------|---------|---------|---------|---------|---------|
| Daily     |         |         |         |         |         |
| Weekday   |         |         |         |         |         |
| Weekend   |         |         |         |         |         |
| Monday    |         |         |         |         |         |
| Tuesday   |         |         |         |         |         |
| Wednesday |         |         |         |         |         |
| Thursday  |         |         |         |         |         |
| Friday    |         |         |         |         |         |
| Saturday  |         |         |         |         |         |
| Sunday    |         |         |         |         |         |

## Service Periods Information Sheets

**CICS subsystem name:** \_\_\_\_\_

| Day       | From-to | From-to | From-to | From-to | From-to |
|-----------|---------|---------|---------|---------|---------|
| Daily     |         |         |         |         |         |
| Weekday   |         |         |         |         |         |
| Weekend   |         |         |         |         |         |
| Monday    |         |         |         |         |         |
| Tuesday   |         |         |         |         |         |
| Wednesday |         |         |         |         |         |
| Thursday  |         |         |         |         |         |
| Friday    |         |         |         |         |         |
| Saturday  |         |         |         |         |         |
| Sunday    |         |         |         |         |         |

**CICS subsystem name:** \_\_\_\_\_

| Day       | From-to | From-to | From-to | From-to | From-to |
|-----------|---------|---------|---------|---------|---------|
| Daily     |         |         |         |         |         |
| Weekday   |         |         |         |         |         |
| Weekend   |         |         |         |         |         |
| Monday    |         |         |         |         |         |
| Tuesday   |         |         |         |         |         |
| Wednesday |         |         |         |         |         |
| Thursday  |         |         |         |         |         |
| Friday    |         |         |         |         |         |
| Saturday  |         |         |         |         |         |
| Sunday    |         |         |         |         |         |

**CICS subsystem name:** \_\_\_\_\_

| Day       | From-to | From-to | From-to | From-to | From-to |
|-----------|---------|---------|---------|---------|---------|
| Daily     |         |         |         |         |         |
| Weekday   |         |         |         |         |         |
| Weekend   |         |         |         |         |         |
| Monday    |         |         |         |         |         |
| Tuesday   |         |         |         |         |         |
| Wednesday |         |         |         |         |         |
| Thursday  |         |         |         |         |         |
| Friday    |         |         |         |         |         |
| Saturday  |         |         |         |         |         |
| Sunday    |         |         |         |         |         |

## Link Monitoring Information Sheets

Read “How to Use Link Monitoring” on page 30 and “CONNECTION - Link monitoring definitions” on page 70 then fill out these worksheets.

**CICS subsystem name:** \_\_\_\_\_

| Link SYSID | ECHO                                                        | Other end<br>(circle one) | UTCDIFF<br>(time<br>deferential) | Description (up to 16 characters) |
|------------|-------------------------------------------------------------|---------------------------|----------------------------------|-----------------------------------|
|            | <input type="checkbox"/> YES<br><input type="checkbox"/> NO | CICS<br>IMS               |                                  |                                   |
|            | <input type="checkbox"/> YES<br><input type="checkbox"/> NO | CICS<br>IMS               |                                  |                                   |
|            | <input type="checkbox"/> YES<br><input type="checkbox"/> NO | CICS<br>IMS               |                                  |                                   |
|            | <input type="checkbox"/> YES<br><input type="checkbox"/> NO | CICS<br>IMS               |                                  |                                   |
|            | <input type="checkbox"/> YES<br><input type="checkbox"/> NO | CICS<br>IMS               |                                  |                                   |

**CICS subsystem name:** \_\_\_\_\_

| Link SYSID | ECHO                                                        | Other end<br>(circle one) | UTCDIFF<br>(time<br>deferential) | Description (up to 16 characters) |
|------------|-------------------------------------------------------------|---------------------------|----------------------------------|-----------------------------------|
|            | <input type="checkbox"/> YES<br><input type="checkbox"/> NO | CICS<br>IMS               |                                  |                                   |
|            | <input type="checkbox"/> YES<br><input type="checkbox"/> NO | CICS<br>IMS               |                                  |                                   |
|            | <input type="checkbox"/> YES<br><input type="checkbox"/> NO | CICS<br>IMS               |                                  |                                   |
|            | <input type="checkbox"/> YES<br><input type="checkbox"/> NO | CICS<br>IMS               |                                  |                                   |
|            | <input type="checkbox"/> YES<br><input type="checkbox"/> NO | CICS<br>IMS               |                                  |                                   |

## Link Monitoring Information Sheets

**CICS subsystem name:** \_\_\_\_\_

| Link SYSID | ECHO              | Other end<br>(circle one) | UTCDIFF<br>(time<br>deferential) | Description (up to 16 characters) |
|------------|-------------------|---------------------------|----------------------------------|-----------------------------------|
|            | ___ YES<br>___ NO | CICS<br>IMS               |                                  |                                   |
|            | ___ YES<br>___ NO | CICS<br>IMS               |                                  |                                   |
|            | ___ YES<br>___ NO | CICS<br>IMS               |                                  |                                   |
|            | ___ YES<br>___ NO | CICS<br>IMS               |                                  |                                   |
|            | ___ YES<br>___ NO | CICS<br>IMS               |                                  |                                   |

**CICS subsystem name:** \_\_\_\_\_

| Link SYSID | ECHO              | Other end<br>(circle one) | UTCDIFF<br>(time<br>deferential) | Description (up to 16 characters) |
|------------|-------------------|---------------------------|----------------------------------|-----------------------------------|
|            | ___ YES<br>___ NO | CICS<br>IMS               |                                  |                                   |
|            | ___ YES<br>___ NO | CICS<br>IMS               |                                  |                                   |
|            | ___ YES<br>___ NO | CICS<br>IMS               |                                  |                                   |
|            | ___ YES<br>___ NO | CICS<br>IMS               |                                  |                                   |
|            | ___ YES<br>___ NO | CICS<br>IMS               |                                  |                                   |

## Link Monitoring Information Sheets

**CICS subsystem name:** \_\_\_\_\_

| Link SYSID | ECHO              | Other end<br>(circle one) | UTCDIFF<br>(time<br>deferential) | Description (up to 16 characters) |
|------------|-------------------|---------------------------|----------------------------------|-----------------------------------|
|            | ___ YES<br>___ NO | CICS<br>IMS               |                                  |                                   |
|            | ___ YES<br>___ NO | CICS<br>IMS               |                                  |                                   |
|            | ___ YES<br>___ NO | CICS<br>IMS               |                                  |                                   |
|            | ___ YES<br>___ NO | CICS<br>IMS               |                                  |                                   |
|            | ___ YES<br>___ NO | CICS<br>IMS               |                                  |                                   |

**CICS subsystem name:** \_\_\_\_\_

| Link SYSID | ECHO              | Other end<br>(circle one) | UTCDIFF<br>(time<br>deferential) | Description (up to 16 characters) |
|------------|-------------------|---------------------------|----------------------------------|-----------------------------------|
|            | ___ YES<br>___ NO | CICS<br>IMS               |                                  |                                   |
|            | ___ YES<br>___ NO | CICS<br>IMS               |                                  |                                   |
|            | ___ YES<br>___ NO | CICS<br>IMS               |                                  |                                   |
|            | ___ YES<br>___ NO | CICS<br>IMS               |                                  |                                   |
|            | ___ YES<br>___ NO | CICS<br>IMS               |                                  |                                   |

## Startup and Shutdown Trigger Information Sheets

---

### Startup and Shutdown Trigger Information Sheets

Up to 28 event names can be used for each subsystem. Read “TRIGGER - Startup and shutdown triggers” on page 85 before filling out this information.

**CICS subsystem name:** \_\_\_\_\_

| Check one                 | SERVICE<br>(yes/no) | Event<br>name | Event<br>name | Event<br>name | Event<br>name | Event<br>name | Event<br>name |
|---------------------------|---------------------|---------------|---------------|---------------|---------------|---------------|---------------|
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |

**CICS subsystem name:** \_\_\_\_\_

| Check one                 | SERVICE<br>(yes/no) | Event<br>name | Event<br>name | Event<br>name | Event<br>name | Event<br>name | Event<br>name |
|---------------------------|---------------------|---------------|---------------|---------------|---------------|---------------|---------------|
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |

## Startup and Shutdown Trigger Information Sheets

**CICS subsystem name:** \_\_\_\_\_

| Check one                 | SERVICE<br>(yes/no) | Event<br>name | Event<br>name | Event<br>name | Event<br>name | Event<br>name | Event<br>name |
|---------------------------|---------------------|---------------|---------------|---------------|---------------|---------------|---------------|
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |

**CICS subsystem name:** \_\_\_\_\_

| Check one                 | SERVICE<br>(yes/no) | Event<br>name | Event<br>name | Event<br>name | Event<br>name | Event<br>name | Event<br>name |
|---------------------------|---------------------|---------------|---------------|---------------|---------------|---------------|---------------|
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |
| __ STARTUP<br>__ SHUTDOWN |                     |               |               |               |               |               |               |

## Startup and Shutdown Trigger Information Sheets

**CICS subsystem name:** \_\_\_\_\_

| Check one                                                             | SERVICE<br>(yes/no) | Event<br>name | Event<br>name | Event<br>name | Event<br>name | Event<br>name | Event<br>name |
|-----------------------------------------------------------------------|---------------------|---------------|---------------|---------------|---------------|---------------|---------------|
| <input type="checkbox"/> STARTUP<br><input type="checkbox"/> SHUTDOWN |                     |               |               |               |               |               |               |
| <input type="checkbox"/> STARTUP<br><input type="checkbox"/> SHUTDOWN |                     |               |               |               |               |               |               |
| <input type="checkbox"/> STARTUP<br><input type="checkbox"/> SHUTDOWN |                     |               |               |               |               |               |               |
| <input type="checkbox"/> STARTUP<br><input type="checkbox"/> SHUTDOWN |                     |               |               |               |               |               |               |
| <input type="checkbox"/> STARTUP<br><input type="checkbox"/> SHUTDOWN |                     |               |               |               |               |               |               |
| <input type="checkbox"/> STARTUP<br><input type="checkbox"/> SHUTDOWN |                     |               |               |               |               |               |               |
| <input type="checkbox"/> STARTUP<br><input type="checkbox"/> SHUTDOWN |                     |               |               |               |               |               |               |
| <input type="checkbox"/> STARTUP<br><input type="checkbox"/> SHUTDOWN |                     |               |               |               |               |               |               |
| <input type="checkbox"/> STARTUP<br><input type="checkbox"/> SHUTDOWN |                     |               |               |               |               |               |               |

**CICS subsystem name:** \_\_\_\_\_

| Check one                                                             | SERVICE<br>(yes/no) | Event<br>name | Event<br>name | Event<br>name | Event<br>name | Event<br>name | Event<br>name |
|-----------------------------------------------------------------------|---------------------|---------------|---------------|---------------|---------------|---------------|---------------|
| <input type="checkbox"/> STARTUP<br><input type="checkbox"/> SHUTDOWN |                     |               |               |               |               |               |               |
| <input type="checkbox"/> STARTUP<br><input type="checkbox"/> SHUTDOWN |                     |               |               |               |               |               |               |
| <input type="checkbox"/> STARTUP<br><input type="checkbox"/> SHUTDOWN |                     |               |               |               |               |               |               |
| <input type="checkbox"/> STARTUP<br><input type="checkbox"/> SHUTDOWN |                     |               |               |               |               |               |               |
| <input type="checkbox"/> STARTUP<br><input type="checkbox"/> SHUTDOWN |                     |               |               |               |               |               |               |
| <input type="checkbox"/> STARTUP<br><input type="checkbox"/> SHUTDOWN |                     |               |               |               |               |               |               |
| <input type="checkbox"/> STARTUP<br><input type="checkbox"/> SHUTDOWN |                     |               |               |               |               |               |               |
| <input type="checkbox"/> STARTUP<br><input type="checkbox"/> SHUTDOWN |                     |               |               |               |               |               |               |
| <input type="checkbox"/> STARTUP<br><input type="checkbox"/> SHUTDOWN |                     |               |               |               |               |               |               |



## Health Checking Information Sheets

Read "How to Use Health Checking" on page 28 before filling out this information.

**CICS subsystem name:** \_\_\_\_\_

| PROGRAM | Description |
|---------|-------------|
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |

**CICS subsystem name:** \_\_\_\_\_

| PROGRAM | Description |
|---------|-------------|
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |

## Health Checking Information Sheets

**CICS subsystem name:** \_\_\_\_\_

| PROGRAM | Description |
|---------|-------------|
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |

**CICS subsystem name:** \_\_\_\_\_

| PROGRAM | Description |
|---------|-------------|
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |

**CICS subsystem name:** \_\_\_\_\_

| PROGRAM | Description |
|---------|-------------|
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |
|         |             |

---

## System IPLs, NetView Restarts, and CICS Restarts

Schedule the following:

## System IPLs, NetView Restarts, and CICS Restarts

| Event           | Description                                                                                                                                                                                                                                                  | Resource name | When |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|------|
| MVS IPL         | Your MVS system will need to be IPLed after the IEAAPFxx member has been changed. You will also need to IPL MVS if you change the subsystem name table (refer to "Step 5: Check the Subsystem Name Table" on page 154). Do this before NetView is restarted. |               |      |
| NetView restart | This restart takes place after the customization steps are complete.                                                                                                                                                                                         |               |      |
| CICS restart    | If a new library was added as a result of the basic CICS definitions, the CICS system needs to be restarted.                                                                                                                                                 |               |      |

## System IPLs, NetView Restarts, and CICS Restarts

| Event           | Description                                                                                                                                                                                                                                                  | Resource name | When |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|------|
| MVS IPL         | Your MVS system will need to be IPLed after the IEAAPFxx member has been changed. You will also need to IPL MVS if you change the subsystem name table (refer to “Step 5: Check the Subsystem Name Table” on page 154). Do this before NetView is restarted. |               |      |
| NetView restart | This restart takes place after the customization steps are complete.                                                                                                                                                                                         |               |      |
| CICS restart    | If a new library was added as a result of the basic CICS definitions, the CICS system needs to be restarted.                                                                                                                                                 |               |      |

## System IPLs, NetView Restarts, and CICS Restarts

| Event           | Description                                                                                                                                                                                                                                                  | Resource name | When |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|------|
| MVS IPL         | Your MVS system will need to be IPLed after the IEAAPFxx member has been changed. You will also need to IPL MVS if you change the subsystem name table (refer to “Step 5: Check the Subsystem Name Table” on page 154). Do this before NetView is restarted. |               |      |
| NetView restart | This restart takes place after the customization steps are complete.                                                                                                                                                                                         |               |      |
| CICS restart    | If a new library was added as a result of the basic CICS definitions, the CICS system needs to be restarted.                                                                                                                                                 |               |      |

## Appendix F. Extra Check Lists

### Planning Check Lists

| √     | Step                                                                                                                                                                                                                                                                                                                   |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| _____ | 1: Verify that SA OS/390 on each NetView domain in which CICS 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 are going to 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: Fill out the worksheets. See "Designing Your Automation Environment" on page 152.                                                                                                                                                                                                                                   |
| _____ | 4: Schedule CICS subsystem initializations, NetView initializations, and system IPLs according to the directions in the worksheet shown in "System IPLs, NetView Restarts, and CICS Restarts" on page 327.                                                                                                             |

| √     | Step                                                                                                                                                                                                                                                                                                                   |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| _____ | 1: Verify that SA OS/390 on each NetView domain in which CICS 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 are going to 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: Fill out the worksheets. See "Designing Your Automation Environment" on page 152.                                                                                                                                                                                                                                   |
| _____ | 4: Schedule CICS subsystem initializations, NetView initializations, and system IPLs according to the directions in the worksheet shown in "System IPLs, NetView Restarts, and CICS Restarts" on page 327.                                                                                                             |

| √     | Step                                                                                                                                                                                                                                                                                                                   |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| _____ | 1: Verify that SA OS/390 on each NetView domain in which CICS 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 are going to 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: Fill out the worksheets. See "Designing Your Automation Environment" on page 152.                                                                                                                                                                                                                                   |
| _____ | 4: Schedule CICS subsystem initializations, NetView initializations, and system IPLs according to the directions in the worksheet shown in "System IPLs, NetView Restarts, and CICS Restarts" on page 327.                                                                                                             |

## Installation Check Lists

### Installation Check Lists

| √     | Step                                                                                                                                                                                                                                                             |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| _____ | 1: Using the information in the CICS Automation program directory, load the CICS Automation libraries from the distribution tape.                                                                                                                                |
| _____ | 2: Add the following entry in MPFLSTxx in SYS1.PARMLIB to trap all DFH and EVE prefix messages:<br><br>DFH*,SUP(NO),AUTO(YES).<br>EVE*,SUP(NO),AUTO(YES).<br>AND*,SUP(NO),AUTO(YES).<br><br>This CICS Automation requirement forwards these messages to NetView. |
| _____ | 3: Update IEAAPFxx in SYS1.PARMLIB to provide APF authorization.                                                                                                                                                                                                 |
| _____ | 4. Define subsystem allocatable consoles.                                                                                                                                                                                                                        |
| _____ | 5. Check the subsystem name table to ensure that the NetView SSI is first.                                                                                                                                                                                       |

| √     | Step                                                                                                                                                                                                                                                             |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| _____ | 1: Using the information in the CICS Automation program directory, load the CICS Automation libraries from the distribution tape.                                                                                                                                |
| _____ | 2: Add the following entry in MPFLSTxx in SYS1.PARMLIB to trap all DFH and EVE prefix messages:<br><br>DFH*,SUP(NO),AUTO(YES).<br>EVE*,SUP(NO),AUTO(YES).<br>AND*,SUP(NO),AUTO(YES).<br><br>This CICS Automation requirement forwards these messages to NetView. |
| _____ | 3: Update IEAAPFxx in SYS1.PARMLIB to provide APF authorization.                                                                                                                                                                                                 |
| _____ | 4. Define subsystem allocatable consoles.                                                                                                                                                                                                                        |
| _____ | 5. Check the subsystem name table to ensure that the NetView SSI is first.                                                                                                                                                                                       |

| √     | Step                                                                                                                                                                                                                                                             |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| _____ | 1: Using the information in the CICS Automation program directory, load the CICS Automation libraries from the distribution tape.                                                                                                                                |
| _____ | 2: Add the following entry in MPFLSTxx in SYS1.PARMLIB to trap all DFH and EVE prefix messages:<br><br>DFH*,SUP(NO),AUTO(YES).<br>EVE*,SUP(NO),AUTO(YES).<br>AND*,SUP(NO),AUTO(YES).<br><br>This CICS Automation requirement forwards these messages to NetView. |
| _____ | 3: Update IEAAPFxx in SYS1.PARMLIB to provide APF authorization.                                                                                                                                                                                                 |
| _____ | 4. Define subsystem allocatable consoles.                                                                                                                                                                                                                        |
| _____ | 5. Check the subsystem name table to ensure that the NetView SSI is first.                                                                                                                                                                                       |



## Merging NetView Related Members Check Lists

| √     | Step                                                         |
|-------|--------------------------------------------------------------|
| _____ | 1: Add CICS Automation data sets to NetView JCL              |
| _____ | 2: Copy CICS Automation sample members to the target library |
| _____ | 3: Merge Status Display Facility members                     |
| _____ | 4: Merge EVECFG into the control file                        |
| _____ | 5: Merge EVECMD into DSICMD                                  |
| _____ | 6: Merge the NetView Automation Table                        |
| _____ | 7: Merge for CDRM recovery                                   |
| _____ | 8: Merge EVEOPF into DSIOPF                                  |
| _____ | 9: Merge the NetView profile data set                        |
| _____ | 10: Merge EVEDMN into DSIDMN                                 |

| √     | Step                                                         |
|-------|--------------------------------------------------------------|
| _____ | 1: Add CICS Automation data sets to NetView JCL              |
| _____ | 2: Copy CICS Automation sample members to the target library |
| _____ | 3: Merge Status Display Facility members                     |
| _____ | 4: Merge EVECFG into the control file                        |
| _____ | 5: Merge EVECMD into DSICMD                                  |
| _____ | 6: Merge the NetView Automation Table                        |
| _____ | 7: Merge for CDRM recovery                                   |
| _____ | 8: Merge EVEOPF into DSIOPF                                  |
| _____ | 9: Merge the NetView profile data set                        |
| _____ | 10: Merge EVEDMN into DSIDMN                                 |

| √     | Step                                                         |
|-------|--------------------------------------------------------------|
| _____ | 1: Add CICS Automation data sets to NetView JCL              |
| _____ | 2: Copy CICS Automation sample members to the target library |
| _____ | 3: Merge Status Display Facility members                     |
| _____ | 4: Merge EVECFG into the control file                        |
| _____ | 5: Merge EVECMD into DSICMD                                  |
| _____ | 6: Merge the NetView Automation Table                        |
| _____ | 7: Merge for CDRM recovery                                   |
| _____ | 8: Merge EVEOPF into DSIOPF                                  |
| _____ | 9: Merge the NetView profile data set                        |
| _____ | 10: Merge EVEDMN into DSIDMN                                 |

---

## CICS Definitions in NetView Check Lists

|       |                                           |
|-------|-------------------------------------------|
| √     | <b>Step</b>                               |
| _____ | 1: SIT or startup overrides               |
| _____ | 2: Program list table definitions         |
| _____ | 3: Transaction list table definitions     |
| _____ | 4: Define consoles                        |
| _____ | 5: Transaction and program definitions    |
| _____ | 6: DFHRPL and the CICS Automation library |

|       |                                           |
|-------|-------------------------------------------|
| √     | <b>Step</b>                               |
| _____ | 1: SIT or startup overrides               |
| _____ | 2: Program list table definitions         |
| _____ | 3: Transaction list table definitions     |
| _____ | 4: Define consoles                        |
| _____ | 5: Transaction and program definitions    |
| _____ | 6: DFHRPL and the CICS Automation library |

|       |                                           |
|-------|-------------------------------------------|
| √     | <b>Step</b>                               |
| _____ | 1: SIT or startup overrides               |
| _____ | 2: Program list table definitions         |
| _____ | 3: Transaction list table definitions     |
| _____ | 4: Define consoles                        |
| _____ | 5: Transaction and program definitions    |
| _____ | 6: DFHRPL and the CICS Automation library |

|       |                                           |
|-------|-------------------------------------------|
| √     | <b>Step</b>                               |
| _____ | 1: SIT or startup overrides               |
| _____ | 2: Program list table definitions         |
| _____ | 3: Transaction list table definitions     |
| _____ | 4: Define consoles                        |
| _____ | 5: Transaction and program definitions    |
| _____ | 6: DFHRPL and the CICS Automation library |

|       |                                           |
|-------|-------------------------------------------|
| √     | <b>Step</b>                               |
| _____ | 1: SIT or startup overrides               |
| _____ | 2: Program list table definitions         |
| _____ | 3: Transaction list table definitions     |
| _____ | 4: Define consoles                        |
| _____ | 5: Transaction and program definitions    |
| _____ | 6: DFHRPL and the CICS Automation library |

## CICS Automation Definitions in NetView Check Lists

|       |                                                                      |
|-------|----------------------------------------------------------------------|
| √     | <b>Step</b>                                                          |
| _____ | 1: Basic CICS automation common control file definitions             |
| _____ | 2: Basic CICS subsystem control file definitions                     |
| _____ | 3: CICS Automation security checking                                 |
| _____ | 4: Extended CICS automation common control file definitions          |
| _____ | 5: Extended CICS subsystem common control file definitions           |
| _____ | 6: The program-to-program interface initialization member (EVENTASK) |
| _____ | 7: The REXX environment table                                        |
| _____ | 8: Customizing the Status Display Facility                           |

|       |                                                                      |
|-------|----------------------------------------------------------------------|
| √     | <b>Step</b>                                                          |
| _____ | 1: Basic CICS automation common control file definitions             |
| _____ | 2: Basic CICS subsystem control file definitions                     |
| _____ | 3: CICS Automation security checking                                 |
| _____ | 4: Extended CICS automation common control file definitions          |
| _____ | 5: Extended CICS subsystem common control file definitions           |
| _____ | 6: The program-to-program interface initialization member (EVENTASK) |
| _____ | 7: The REXX environment table                                        |
| _____ | 8: Customizing the Status Display Facility                           |

|       |                                                                      |
|-------|----------------------------------------------------------------------|
| √     | <b>Step</b>                                                          |
| _____ | 1: Basic CICS automation common control file definitions             |
| _____ | 2: Basic CICS subsystem control file definitions                     |
| _____ | 3: CICS Automation security checking                                 |
| _____ | 4: Extended CICS automation common control file definitions          |
| _____ | 5: Extended CICS subsystem common control file definitions           |
| _____ | 6: The program-to-program interface initialization member (EVENTASK) |
| _____ | 7: The REXX environment table                                        |
| _____ | 8: Customizing the Status Display Facility                           |

|       |                                                                      |
|-------|----------------------------------------------------------------------|
| √     | <b>Step</b>                                                          |
| _____ | 1: Basic CICS automation common control file definitions             |
| _____ | 2: Basic CICS subsystem control file definitions                     |
| _____ | 3: CICS Automation security checking                                 |
| _____ | 4: Extended CICS automation common control file definitions          |
| _____ | 5: Extended CICS subsystem common control file definitions           |
| _____ | 6: The program-to-program interface initialization member (EVENTASK) |
| _____ | 7: The REXX environment table                                        |
| _____ | 8: Customizing the Status Display Facility                           |

## Extended CICS Definitions Check List

---

### Extended CICS Definitions Check List

|       |                                                              |
|-------|--------------------------------------------------------------|
| √     | <b>Step</b>                                                  |
| _____ | 1: Health check program and transaction definitions          |
| _____ | 2: Program-to-program interface initialization member (CICS) |
| _____ | 3: Add or change the CICS transient data messages            |
| _____ | 4: Echoplex back end programs                                |
| _____ | 5: Security considerations                                   |

|       |                                                              |
|-------|--------------------------------------------------------------|
| √     | <b>Step</b>                                                  |
| _____ | 1: Health check program and transaction definitions          |
| _____ | 2: Program-to-program interface initialization member (CICS) |
| _____ | 3: Add or change the CICS transient data messages            |
| _____ | 4: Echoplex back end programs                                |
| _____ | 5: Security considerations                                   |

|       |                                                              |
|-------|--------------------------------------------------------------|
| √     | <b>Step</b>                                                  |
| _____ | 1: Health check program and transaction definitions          |
| _____ | 2: Program-to-program interface initialization member (CICS) |
| _____ | 3: Add or change the CICS transient data messages            |
| _____ | 4: Echoplex back end programs                                |
| _____ | 5: Security considerations                                   |

|       |                                                              |
|-------|--------------------------------------------------------------|
| √     | <b>Step</b>                                                  |
| _____ | 1: Health check program and transaction definitions          |
| _____ | 2: Program-to-program interface initialization member (CICS) |
| _____ | 3: Add or change the CICS transient data messages            |
| _____ | 4: Echoplex back end programs                                |
| _____ | 5: Security considerations                                   |

|       |                                                              |
|-------|--------------------------------------------------------------|
| √     | <b>Step</b>                                                  |
| _____ | 1: Health check program and transaction definitions          |
| _____ | 2: Program-to-program interface initialization member (CICS) |
| _____ | 3: Add or change the CICS transient data messages            |
| _____ | 4: Echoplex back end programs                                |
| _____ | 5: Security considerations                                   |

---

## Glossary of CICS and Other Terms

This glossary defines special CICS 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 gives the particular sense in which it is used in the SA OS/390 CICS Automation library.

**abend.** Abnormal end of task.

**ACB.** Access method control block (VTAM and VSAM).

**ACK.** Acknowledgement.

**APAR.** Authorized program analysis report.

**application program.** (1) A program written for or by a user that applies to the user's work. (2) In data communication, a program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

**ASCII.** American National Standard Code for Information Interchange.

**batch.** An accumulation of data to be processed.

**CEC.** Central Electronic Complex.

**CEMT.** The CICS master terminal transaction.

**central electronic complex (CEC).** 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 4-processor CEC, or can be partitioned into the equivalent of two 3090 model 200s, each of which runs as a CEC with its own operating system.

**CICS.** Customer Information Control System.

**command.** In CICS, an instruction similar in format to a high-level programming language statement. (Contrast with macro.) CICS commands invariably include the verb EXECUTE (or EXEC). They may be issued by an application program to make use of CICS facilities.

**command-language statement.** In CICS, synonym for command.

**concurrent.** Pertaining to the occurrence of two or more activities within a given interval of time.

**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.

**end user.** In CICS, anyone using CICS 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.

**initial program load (IPL).** The initialization procedure that causes an operating system to commence operation.

**initialization.** (1) Actions performed by CICS to construct the environment in the CICS region to enable CICS applications to be run. (2) A process started by SA OS/390 and CICS Automation to construct the environment in which automation is to occur.

**installation.** (1) 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. (2) The task of making a program ready to do useful work. This task includes generating a program, initializing it, and applying PTFs to it.

**intercommunication facilities.** A generic term covering intersystem communication (ISC) and multiregion operation (MRO).

**interregion communication (IRC).** The method by which CICS provides communication between a CICS region and another region in the same processor. Used for multiregion operation.

**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 CICS systems, and may be used for user application to user application communication, or for transparently executing CICS functions on a remote CICS system.

**IPL.** Initial Program Load.

**IRC.** Interregion communication.

**ISC.** Intersystem communication.

**keyword.** (1) A symbol that identifies a parameter. (2) A part of a command operand that consists of a specific character string. (3) An operand in a CEDD definition. Key-sequenced data set—a VSAM data base organization.

**local.** In data communication, pertaining to devices that are attached to a controlling unit 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 or control unit is used. Contrast with remote device.

**local system.** In CICS intercommunication, the CICS system from whose point of view intercommunication is being discussed.

**NACK.** Negative Acknowledgement.

**network.** (1) An interconnected group of nodes. (2) 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.

**online.** (1) Pertaining to a user's ability to interact with a computer. (2) Pertaining to a user's access to a computer via a terminal.

**operating system.** Software that controls the execution of programs; an operating system may provide services such as resource allocation, scheduling, input/output control, and data management.

**parameter.** (ISO) A variable that is given a constant value for a specified application and that may denote the application. A unique string of characters that a program, computer

**processor.** (ISO) In a computer, a functional unit that interprets and executes instructions.

**program temporary fix (PTF).** A temporary solution or by-pass of a problem diagnosed by IBM field engineering as the result of a defect in a current unaltered release of the program.

**PTF.** Program Temporary Fix.

**PUT.** Program update tape.

**recovery routine.** A routine that is 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 CICS intercommunication, a system that the local CICS system accesses via intersystem communication or multiregion operation.

**resource.** Any facility of the computing system or operating system required by a job or task, and including main storage, input/output devices, the processing unit, data sets, and control or processing programs.

**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.

**SIT.** System Initialization Table.

**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 CICS by the system operator.

**subsystem.** (1) A secondary or subordinate system. (2) A resource defined to SA OS/390 and CICS Automation.

**system.** In CICS, an assembly of hardware and software capable of providing the facilities of CICS for a particular installation.

**system initialization table (SIT).** 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.** (1) (ISO) A basic unit of work to be accomplished by a computer. (2) Under CICS, the execution of a transaction for a particular user.

**terminal.** (1) A point in a system or communication network at which data can either enter or leave. (2) In

CICS, 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.

**update.** To modify a file with current information.

**VTAM.** An acronym for the Virtual Telecommunications Access Method. This is one of the ways CICS communicates with terminals.





---

# Index

## Special Characters

%INCLUDE - Include additional members 46  
%INCLUDE statement for CICS Automation control file entries 229

## A

ABCODESYSTEM - System abend recovery 60  
ABCODETRAN - Transaction abend recovery 62, 103  
ACK response 200  
ACORESTART - NetView restart commands 47  
Add or change the CICS transient data messages 192  
Allocatable consoles, defining 154  
APF authorization 153  
Application health 8  
AREA - Define a set of State/Action Tables 64  
Autoinstall 15, 52  
Autoinstall problem recovery State/Action Table 287  
Autoinstall time limit 79  
Automated operator profile 158  
Automatic installation of terminals 9  
Automation flags 48  
Automation operator profiles 291  
Automation operators 4, 50  
Automation table 157  
AUTOOPS - Automation operators 50

## B

Back end programs, echoplex 193  
Basic CICS automation common control file definitions 183  
Basic CICS subsystem control file definitions 184  
Build message processor 89

## C

CANCEL from NetView 202  
CDRM recovery 157  
CEMT 9  
CEMT PPI short syntax 109  
CEMT support for timeouts 54  
CEMT PPI - CEMT PPI short syntax 109  
Change the shutdown delay time 110  
checking  
    health 29  
checklist 151  
CICS Automation control file entries 229  
CICS Automation library 182  
CICS Automation NetView Automation Table entries 244, 247

CICS automation operator profiles 291  
CICS Automation security checking 184  
CICS CEMT support for timeouts 54  
CICS dump 117  
CICS error thresholds 58  
CICS extensions for SUBSYSTEM entry 66  
CICS message router 118  
CICS PPI initialization member 139  
CICS receiver program 199  
CICS region status 8  
CICS requests using the program-to-program interface 203  
CICS resync 113  
CICS system information 9  
CICS to NetView communication interface 213  
CICS Transactions. 15, 52  
CICS transient data messages 192  
CICS, echoplexing to 30  
CICSBMSG - Build message processor 89  
CICSCNTL - CICS extensions for SUBSYSTEM entry 66  
CICSDLY - Change the shutdown delay time 110  
CICSGROUP - Group of CICS subsystems 69  
CICSHLTH - linemode health checking 122  
CICSLM - linemode link monitor 130  
CICSMHDR - linemode message options 133  
CICSMSG0 157  
CICSMSG0 - Early NetView Initialization Automation Table 244  
CICSMSG1 157  
CICSMSG1 - Production NetView Initialization Automation Table 247  
CICSOVRD - linemode SIT override 128  
CICSplex SM Address Space (CMAS) shutdown 137  
CICSplex SM REXX API  
    installing 190  
CICSPOST - Post an external event to a specific subsystem 111  
CICSPURG - Purge transactions 112  
CICSQRY - Name lookup 91  
CICSRCMD - Request a CICS function 95  
CICSRSYC - CICS resync 113  
CICSSEC - Invoke security checking 96  
CICSSHUT - Shutdown processor 114  
CICSSTOP - linemode CICS shutdown 126  
CICSSTRT - linemode CICS startup 124  
CICSTRCE - linemode trace options 135  
CMASSHUT - CICSplex SM Address Space (CMAS) shutdown 137  
Command Model statements (building member) 156  
Command model statements member CICS Automation entries 277

- Common response handler from CICS 212
- CONNECTION - Link monitoring definitions 70
- Console defining (CICS) 161
- control file 5, 156
  - Entry notation and syntax rules 44
- control file definitions 183
- control file entries
  - %INCLUDE - Include additional members 46
  - ABCODESYSTM - System abend recovery 60
  - ABCODETRAN - Transaction abend recovery 62
  - ACORESTART - NetView restart commands 47
  - AREA - Define a set of State/Action Tables 64
  - Autoinstall time limit 79
  - Automation flags 48
  - Automation operators 50
  - AUTOOPS - Automation operators 50
  - CEMT support for timeouts 54
  - CICS CEMT support for timeouts 54
  - CICS error thresholds 58
  - CICS extensions for SUBSYSTEM entry 66
  - CICS initialization support 53
  - CICSCNTL - CICS extensions for SUBSYSTEM entry 66
  - CICSGROUP - Group of CICS subsystems 69
  - CONNECTION - Link monitoring definitions 70
  - control file, additional members 46
  - ENVIRON SETUP - CICS initialization support 53
  - ENVIRON TIMEOUT - CICS CEMT support 54
  - EXTCOND - External conditions 73
  - Extensions for SUBSYSTEM entry 66
  - External conditions 73
  - Group of CICS subsystems 69
  - Health checking 75
  - HEALTHCHK - Health checking 75
  - Include additional members 46
  - Initialization setup support 53
  - INITSTART - Automation flag 48
  - Link monitoring definitions 70
  - LISTSHUT - Transaction purging during shutdown 76
  - LM definitions 70
  - NetView restart commands 47
  - Product 64, 77
  - PRODUCT - Subsystem State/Action Table sets 77
  - RCVRAUTO - Autoinstall time limit 79
  - RCVRSOS - Short-on-storage handler 80
  - RCVRTRAN - Transaction recovery 81
  - RCVRVIOL - Storage violation handler 82
  - RECOVERY - Automation flag 48
  - RESTART - Automation flag 48
  - Restart commands 47
  - SERVICE - Service period definitions 83
  - Service period definitions 83
  - Short-on-storage handler 80
  - Shutdown control file entries and replies 55
  - Shutdown replies 55

- control file entries (*continued*)
  - STARTUP - Startup commands 56
  - Startup and shutdown triggers 85
  - Startup commands 56
  - Status Display Facility support 52
  - Storage violation handler 82
  - SUBSYSTEM - Subsystem definition 57
  - Subsystem definition 57
  - System abend recovery 60
  - THRESHOLDS - CICS error thresholds 58
  - Transaction abend recovery 62
  - Transaction purging during shutdown 76
  - Transaction recovery 81
  - TRIGGER - Startup and shutdown triggers 85
  - Trigger conditions 73
  - Trigger definitions 73
- Control file entry notation 44
- control file, additional members 46
- CONVERSE from CICS 203
- CONVERSE from NetView 200
- COPC 7
- Critical message handler for the Status Display Facility 116
- Critical messages 15, 52
- Current executing domain 8
- Customizing the Status Display Facility 187

## D

- Data control functions 9
- defining
  - Allocatable consoles 154
  - NetView SSI 154
  - SSI, NetView 154
  - Subsystem allocatable consoles 154
  - Subsystem name table 154
- DFHCSD 161
- DFHPLT 160
- DFHRPL, CICS Automation library 182
- DFHSIT 159
- DFHXLTL 160
- Domain member CICS Automation entries 280
- DSIDMN 155, 158, 183
- DSIOPF 158
- DSIPARM 155, 156, 183
- DSIPRF 158
- dump 117
- Dump control functions 9

## E

- Echoplex back end programs 193
- Echoplexing backend programs 12
- Echoplexing, how it works 30
- ENVIRON SETUP - CICS initialization support 53

ENVIRON TIMEOUT - CICS CEMT support 54  
 environment blocks 187  
 Environment table 187  
 EVECAREA - Layout of the health checking  
   DFHCOMMAREA 295  
 EVECFG 156  
 EVECFG - %INCLUDE statement for CICS Automation  
   control file entries 229  
 EVECFG01 - Sample CICS Automation control file  
   entries 229  
 EVECHLTH - Health check sample program (COBOL  
   version) 292  
 EVECMD - NetView command model statements  
   member CICS Automation entries 277  
 EVEDMN - NetView domain member CICS Automation  
   entries 280  
 EVEED003 - Critical message handler for the Status  
   Display Facility 116  
 EVEEMIGR - Migrate subsystem to CICS/TS 119  
 EVEERDMP - CICS dump 117  
 EVEESA01 - Short-on-storage State/Action Table 284  
 EVEESA02 - Storage violation State/Action Table 286  
 EVEESA03 - Autoinstall problem recovery State/Action  
   Table 287  
 EVEESA04 - VTAM ACB problem recovery State/Action  
   Table 288  
 EVEEY00S - Common state handler for State/Action  
   Tables 120  
 EVEEZ000 158  
 EVEFOPF 158  
 EVEMCON0 157  
 EVEMCON0 - Messages in Conflict with AOC/MVS  
   (Included by CICSMSG0) 246  
 EVEMCON1 157  
 EVEMCON1 - Messages in Conflict with AOC/MVS  
   (Included by CICSMSG1) 264  
 EVEMDFH1 157  
 EVEMDFH1 - DFH Messages (Included by  
   CICSMSG1) 250  
 EVEMEVE0 157  
 EVEMEVE0 - EVE Messages (Included by  
   CICSMSG0) 245  
 EVEMEVE1 157  
 EVEMEVE1 - EVE Messages (Included by  
   CICSMSG1) 248  
 EVEMEYU1 - CPSM Messages (Included by  
   EVEMSG01) 302  
 EVEMPINT - EVESCCCI parameter list copy book 216  
 EVEMSG00 157  
 EVEMSG01 157  
 EVENTASK 6  
 EVENTASK - NetView PPI initialization member 141  
 EVENTASK - PPI initialization member (NetView) 296  
 EVEOPF 158  
 EVEOPF - NetView operator definitions CICS  
   Automation entries 281  
 EVEPNLS 156  
 EVEPNLS - The Status Display Facility panels 283  
 EVEPRFAO 158  
 EVEPRFAO - CICS automation operator profiles 291  
 EVESCCCI - CICS to NetView communication  
   interface 213  
 EVESCCCI parameter list copy book 216  
 EVESCMT2 - Message exit table for CICS Version  
   2 267  
 EVESCMT2-Message exit table for CICS Version  
   2 143  
 EVESCMT3 - Message exit table for CICS Version 3  
   and higher 145, 271  
 EVESECUR - Security definitions 279  
 EVESJ010 (job) 221  
 EVESJ015 (job) 223  
 EVESJ020 (job) 224  
 EVESJ021 (job) 187  
 EVESJ022 (job) 226  
 EVESNCCI 6, 200  
 EVESNCCI - NetView to CICS communication  
   interface 207  
 EVESNPPI 199  
 EVESNRSP 6  
 EVESNRSP - Common response handler from  
   CICS 212  
 EVESPINM - PPI initialization member (CICS) 298  
 EVESPINM-CICS PPI initialization member 139  
 EVESPPIC 199  
 EVESROUT - CICS message router 118  
 EVESTS - Status file interface 99  
 EVETREE 156  
 EVETREE - The Status Display Facility tree  
   structure 282  
 EXTCOND - External conditions 73  
 Extended CICS automation common control file  
   definitions 186  
 Extended CICS subsystem common control file  
   definitions 186  
 Extensions for SUBSYSTEM entry 66  
 External conditions 73  
 External events 111

**F**  
 Focal point 12  
 Free-format CEMT requests 9

**G**  
 Group of CICS subsystems 69  
 Grouping CICS 22

## H

Health check program and transaction definitions 191  
Health check sample program (COBOL version) 292  
Health checking 8, 15, 28, 29, 52, 75  
HEALTHCHK - Health checking 75  
Hierarchy 18

## I

IEAAPFxx in SYS1.PARMLIB, updating 153  
IMS, echoplexing to 30  
Include additional members 46  
Initialization member (CICS) 139, 191  
Initialization member (EVENTASK) 187  
Initialization setup support 53  
INITSTART - Automation flag 48  
Installation 153  
Inter-CICS connection management 9  
Invoke security checking 96

## J

Journal control functions 9

## L

Last start type performed 8  
Layout of the health checking DFHCOMMAREA 295  
linemode CICS shutdown 126  
linemode CICS startup 124  
linemode health checking 122  
linemode link monitor 130  
linemode message options 133  
linemode SIT override 128  
linemode trace options 135  
Link monitoring 15, 30, 52  
Link monitoring definitions 70  
links  
    Operator interface 8  
LISTSHUT - Transaction purging during shutdown 76  
LM definitions 70  
Local application support 33  
Local Functions 33

## M

Merging the control file 156  
Message exit table for CICS Version 2 143, 267  
Message exit table for CICS Version 3 and higher 145, 271  
Message router 118  
Messages, critical 15, 52  
Migrate subsystem to CICS/TS 119

## N

NACK response 200  
Name lookup 91  
NetView Automation Table 4, 157  
NetView command model statements member CICS Automation entries 277  
NetView domain member CICS Automation entries 280  
NetView operator definitions CICS Automation entries 281  
NetView PPI initialization member 141  
NetView profile data set 158  
NetView requests using the program-to-program interface 200  
NetView restart commands 47  
NetView subtask program 199  
NetView to CICS communication interface 207

## O

Operator definition member 158  
Operator definitions CICS Automation entries 281  
Operator interface 7

## P

panel  
    Application health 8  
    Broadcast messages 9  
    CEMT 9  
    CICS Monitor 17  
    Health checking 8, 29  
    Messages 18  
    Overview 8  
    Program-to-program interface 10  
    Service periods 9  
    Shutdown 8  
    Startup 8  
    Status Display facility 16  
    Triggers 9  
Panel support 33  
Parameter data sets 155, 183  
Parent/child lookup table 18  
Parent/child relationships 18  
Planning 151  
Post an external event to a specific subsystem 111  
PPI initialization member 141  
Previous region start time 8  
Product 64, 77  
PRODUCT - Subsystem State/Action Table sets 77  
Profile data set 158  
Profile data sets 155, 183  
Profile, automated operators 158  
Program control functions 9

- Program definitions (CICS) 161
- Program list table definitions 160
- Program-to-program interface initialization member 141
- Program-to-program interface initialization member (CICS) 191
- Program-to-program interface initialization member (EVENTASK) 187
- Purge transactions 112
- Purging transactions (shutdown) 76

## R

- RCVRAUTO - Autoinstall time limit 79
- RCVRSOS - Short-on-storage handler 80
- RCVRTRAN - Transaction recovery 81
- RCVRVIOL - Storage violation handler 82
- RECOVERY - Automation flag 48
- Recovery of transactions 23
- Remote Site Recovery for VSAM RLS
  - How to Implement 37
- Request a CICS function 95
- Response handler from CICS 212
- RESPONSE response 200
- RESTART - Automation flag 48
- Restart commands 47
- Resync, CICS 113
- REXX environment blocks 187
- REXX environment table 187

## S

- SA OS/390
  - Status Display Facility 7
- SA OS/390 control file 5
- SA OS/390 status file 5
- scheduling
  - shutdowns 9
  - startups 9
- SDF 52
- Security checking 31, 96, 112, 184
- Security definitions 279
- Segment support in NetView 208
- SEND from CICS 204
- SEND from NetView 202
- Sequencing shutdowns 18
- Sequencing startups 18
- SERVICE - Service period definitions 83
- Service period definitions 83
- service periods
  - overview 9
- Short on storage 15, 52
- Short-on-storage handler 80
- Short-on-storage State/Action Table 284
- shutdown
  - scheduling 9

- Shutdown control file entries and replies 55
- Shutdown delay time change 110
- Shutdown processor 114
- Shutdown replies 55
- Shutdown sequence 18
- Shutdowns 18
- Single-point-of-control 7, 12
- SIT or startup overrides 159
- SSI, NetView 154
- startup
  - scheduling 9
- STARTUP - Startup commands 56
- Startup and shutdown triggers 85
- Startup commands 56
- Startup sequence 18
- Startups 18
- State/Action Tables 6
- Status Display Facility 187
- Status Display Facility installation. 156
- Status Display Facility panels 283
- Status Display Facility support 52
- Status Display Facility tree structure 282
- Status file 5
- Status file interface 99
- Storage violation handler 82
- Storage violation State/Action Table 286
- Storage violations 15, 52
- SUBSYSTEM - Subsystem definition 57
- Subsystem allocatable consoles, defining 154
- Subsystem definition 57
- Subsystem name table 154
- System abend recovery 60
- System hierarchy 18

## T

- Task control functions 9
- Terminal control functions 9
- THRESHOLDS - CICS error thresholds 58
- Thresholds for transaction recovery 23
- Timers 15, 52
- Trace control functions 9
- Transaction abend recovery 62
- Transaction control functions 9
- Transaction definitions (CICS) 161
- Transaction definitions (health check) 191
- Transaction list table definitions 160
- Transaction purging during shutdown 76
- Transaction recovery 23, 81
- Transaction recovery thresholds 23
- Transient data queue functions 9
- TRIGGER - Startup and shutdown triggers 85
- Trigger conditions 73
- Trigger definitions 73
- triggers
  - overview 9

## **V**

VTAM ACB problem recovery State/Action Table 288

VTAM ACM errors 15, 52

VTAM attributes 8

## **W**

Worksheets 307

## **X**

XLT table 160



---

# Communicating Your Comments to IBM

System Automation for OS/390  
AOC/MVS CICS Automation  
Programmer's Reference  
and Installation Guide  
Version 1 Release 4  
Publication No. SC23-3814-02

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 CICS Automation  
Programmer's Reference  
and Installation Guide  
Version 1 Release 4

Publication No. SC23-3814-02

Overall, how satisfied are you with the information in this book?

|                      | Very Satisfied           | Satisfied                | Neutral                  | Dissatisfied             | Very Dissatisfied        |
|----------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| Overall satisfaction | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

How satisfied are you that the information in this book is:

|                          | Very Satisfied           | Satisfied                | Neutral                  | Dissatisfied             | Very Dissatisfied        |
|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| Accurate                 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Complete                 | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Easy to find             | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Easy to understand       | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Well organized           | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Applicable to your tasks | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

\_\_\_\_\_  
Name

\_\_\_\_\_  
Address

\_\_\_\_\_  
Company or Organization

\_\_\_\_\_

\_\_\_\_\_  
Phone No.

\_\_\_\_\_



Cut or Fold  
Along Line

Fold and Tape

Please do not staple

Fold and Tape



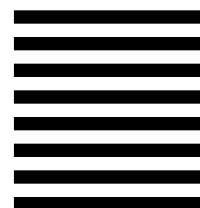
## BUSINESS REPLY MAIL

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

NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES



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.

SC23-3814-02

