



VisualAge Generator Server Guide for MVS, VSE, and VM

Version 1.2



VisualAge Generator Server Guide for MVS, VSE, and VM

Version 1.2

Note

Before using this document, read the general information under "Notices" on page xv.

First Edition (June 1998)

This edition applies to the following licensed programs:

- IBM VisualAge Generator Developer for OS/2 and Windows NT Version 3.1
- IBM VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX Version 3.1
- IBM VisualGen Host Services for OS/400 Version 3.1
- IBM VisualGen Host Services for OS/400 Version 3.6
- IBM VisualAge Generator Server for MVS, VSE, and VM Version 1.2

Order publications by phone or fax. IBM Software Manufacturing Solutions takes publication orders between 8:30 a.m. and 7:00 p.m. eastern standard time (EST). The phone number is (800) 879-2755. The fax number is (800) 284-4721.

You can also order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

A form for comments appears at the back of this publication. If the form has been removed, address your comments to the IBM Corporation, Attn: Information Development, Department T99B Building 062, P.O. Box 12195, Research Triangle Park, NC 27709-2195.

You can fax comments to (919) 254-0206.

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 1994, 1998. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	xv
Trademarks	xvii
Terminology Used in This Document	xviii
About This Document.	xxi
Who Should Use This Document	xxi
Documentation provided with VisualAge Generator	xxii
Related information	xxiii

Part 1. Preparing to Install on MVS, VSE and VM Systems 1

Chapter 1. Preparing for the Installation of VisualAge Generator Server for MVS, VSE, and VM	3
---	----------

Chapter 2. Hardware Requirements for VisualAge Generator Server for MVS, VSE, and VM	5
---	----------

Chapter 3. Software Requirements for VisualAge Generator Server for MVS, VSE, and VM	7
VisualAge Generator Server for MVS	7
VisualAge Generator Server for VSE	9
VisualAge Generator Server for VM	10

Chapter 4. Storage Requirements for VisualAge Generator Server for MVS, VSE, and VM	13
Virtual Storage Requirements	13
VisualAge Generator Server for MVS, VSE, and VM Load Module Storage	13
Application Load Module Storage	14
COBOL Dynamic Storage	15
VisualAge Generator Server for MVS, VSE, and VM Dynamic Storage	16
Storage Requirements for MVS CICS and VSE CICS	17
Disk Storage Requirements for VisualAge Generator Server for MVS, VSE, and VM Work Database Space For Segmented Applications	18

Chapter 5. Installation Considerations for MVS, VSE, and VM Systems	21
Installation and Language-Dependent Options for MVS, VSE, and VM.	21
MVS Batch and MVS/TSO Installation Considerations	24
DL/I Considerations	24
DB2 Considerations	25
Security Considerations	25
Making VisualAge Generator Server for MVS, VSE, and VM, COBOL, and the Generated Modules Available	25
MVS CICS Installation Considerations.	26
CICS/ESA Considerations.	26
DL/I Considerations	26
DB2 Considerations	26
Security Considerations	27
Monitoring and Tuning	27
MVS CICS Utilities	27
Client/Server Processing Considerations	27
Using the /DATA Generation Option	28
Modifying MVS CICS Resource Tables	28
IMS Installation Considerations	29
IMS/ESA Exploitation	29
DB2 Considerations	30
Security Considerations	30
Monitoring and Tuning	30
IMS System Definition	31
IMS Control Region	31
Work Database	31
VSE Batch Installation Considerations.	32
DL/I Considerations	32
DB2/VSE Considerations	33
Making VisualAge Generator Server for MVS, VSE, and VM, COBOL, and the Generated Modules Available	33
VSE CICS Installation Considerations	33
DL/I Considerations	34
DB2/VSE Considerations	34
Security Considerations	34
Monitoring and Tuning	35
VSE CICS Utilities	35

Client/Server Processing Considerations	35	Processing a DISPLAY Process Option for a Printer Map	57
Using the /DATA=24 Generation Option	35	Specifying Print Destination	58
Modifying VSE CICS Resource Tables	36	MVS CICS Terminal Printing	58
VM Batch and VM CMS Installation		CICS Entries for FZETPRT (DBCS only)	62
Considerations	37	Using the New Copy Function	63
SQL/DS Considerations	37	Recovery and Database Integrity	
Security Considerations	37	Considerations	63
Chapter 6. Customizing VisualAge		Specifying Recovery Options in the CICS	
Generator Server for MVS, VSE, and VM	39	Tables	64
General Customization Considerations for		Determining When Database Changes	
MVS, VSE, and VM Systems	39	Are Committed	64
Security Considerations	39	Considerations that Affect Performance	65
Performance Considerations	40	Residency (Modules in Memory)	
Customizing JCL Procedures	41	Considerations	65
Modifying the COBOL Working Storage		Work Database Temporary Storage	
Initialization Option.	41	Queue Considerations	67
Using Generated Programs with PL/I		Database Temporary Storage Size	68
Programs	41	Conversational and	
		Pseudoconversational Processing	68
		Terminal Printing	69
		Monitoring and Tuning the System.	69
		Using and Allocating Data Files in MVS	
		CICS and VSE CICS.	69
		Defining and Loading VSAM Data Files	69
		Using Remote Files	71
		Defining Transient Data Queues.	72
		Considerations for Using DB2 in MVS CICS	73
		Associating DB2 Databases with CICS	
		Transactions	74
		Recovery and Database Integrity	
		Considerations	74
		Considerations for Using DL/I in MVS	
		CICS	74
		Recovery and Database Integrity	
		Considerations	75
		Chapter 9. System Considerations for	
		MVS/TSO	77
		Required File Descriptions	77
		Printing Considerations for MVS/TSO	78
		Making VisualAge Generator Server,	
		COBOL, and the Generated Modules	
		Available	78
		Recovery Considerations	79
		Using VSAM Program Data Files in	
		MVS/TSO	80
		Considerations for Using DB2 in MVS/TSO	80
		Recovery and Database Integrity	
		Considerations	81

Considerations for Using DL/I in MVS/TSO	81
DL/I Support in Generated Programs	81
Defining the Program Specification Block (PSB).	81
Recovery and Database Integrity Considerations	82

Chapter 10. System Considerations for MVS Batch	83
Required File Descriptions	83
Recovery Considerations	83
Using VSAM Program Data Files in MVS Batch.	84
Considerations for Using DB2 in MVS Batch Recovery and Database Integrity Considerations	84
Considerations for Using DL/I in MVS Batch.	85
Defining the Program Specification Block (PSB).	85
Recovery and Database Integrity Considerations	85
Performance Considerations for MVS Batch.	86
Runtime JCL	86

Chapter 11. System Considerations for IMS	87
Required File Descriptions	87
Defining the Program Specification Block (PSB).	88
Migrating from CSP/370RS V1R1	90
Processing Modes	90
Printing Considerations for IMS.	90
Recovery and Database Integrity Considerations	90
Considerations that Affect Performance Residency Considerations and the IMS Preload Function.	92
Database Performance	94
Limiting MFS Control Blocks.	95
Monitoring and Tuning the IMS System	95
Considerations for Using DB2 in IMS	96
Recovery and Database Integrity Considerations	96
Checking Authorization	96
Considerations for Using DL/I in IMS	97
Recovery and Database Integrity Considerations	97

Maintaining the Work Database in IMS	98
Deleting Old Records from the Work Database	98
Expanding the Work Database	101
Supporting Multiple Work Databases	103
Considerations for Message Format Services in IMS	105

Part 3. Preparing and Running Generated Applications on MVS Systems. 111

Chapter 12. Output of Program Generation on MVS Systems	113
List of Program Preparation Steps after Program Generation.	113
Outputs of Generation	114
Objects Generated for Programs.	117
Objects Generated for Tables	119
Objects Generated for Map Groups.	119
Objects Generated for All Part Types	121

Chapter 13. Preparing Generated Programs on MVS Systems.	123
Allocating Preparation Data Sets	123
Running the Preparation JCL.	126
Analyzing the Preparation Listing	126
Analyzing the DB2 Precompile Step	127
Analyzing the CICS Translator Step	127
Analyzing the Compile Step	127
Analyzing the Link-Edit Step.	127
Analyzing the BIND Step	128
Preparing to Run a DB2 Program	129
Granting Authority to Program Users Defining Synonyms for DB2 Tables.	129
Using LE Runtime Options for MVS/TSO and MVS Batch	130

Chapter 14. Creating or Modifying Runtime JCL or CLISTS on MVS Systems	135
Tailoring JCL or CLIST Templates before Generation	135
Modifying Runtime JCL	137
Modifying Runtime CLISTS	139

Chapter 15. Preparing and Running a Generated Program in MVS CICS.	143
Modifying CICS Resource Tables	143
Program Entries (PPT)	144

Transaction Entries (PCT)	145	Chapter 18. Preparing and Running	
Destination Control Table Entries (DCT)	147	Generated Programs in IMS/VS and IMS	
File Control Table Entries (FCT)	147	BMP	165
Resource Control Table Entry (RCT)	148	Modifying the IMS System Definition	
Using Remote Programs, Transactions, or		Parameters	165
Files	148	Defining an Interactive Program	166
Using MVS JES Spool Files	148	Defining Parameters for a Batch Program	
Modifying CICS Startup JCL	148	as an MPP	167
Making New Modules Available in the MVS		Defining Parameters for a Batch-Oriented	
CICS Environment	148	BMP Program	167
Making Programs Resident	149	Defining Parameters for a	
Running Programs under MVS CICS	149	Transaction-Oriented BMP Program	168
Starting the Transaction in MVS CICS	150	Creating MFS Control Blocks.	168
Controlling Diagnostic Information in the		Making New Modules Available in the IMS	
CICS Environment	150	Environment	169
Printing Diagnostic Messages in the		Preloading Program, Print Services, and	
CICS Environment	150	Table Modules	169
Chapter 16. Preparing and Running a		Running Programs under IMS	170
Generated Program in MVS/TSO	151	Starting a Main Program Directly	170
Making VisualAge Generator Server,		Starting a Main Transaction Program	
COBOL, and the Generated Modules		Using the /FORMAT Command	170
Available	151	Running Transaction Programs as IMS	
Running Main Programs under MVS/TSO	152	MPPs	171
Starting the CLIST	152	Running Batch Programs as MPPs	173
Examples of Runtime CLISTs for MVS/TSO	153	Running a Main Program under IMS BMP	173
Running a Main Program with No		Examples of Runtime JCL for IMS BMP	
Database Access	153	Programs	174
Running a Main Program with DB2		Running a Main Batch Program as an	
Access	154	IMS BMP Program	174
Running a Main Program with DL/I		Running a Main Batch Program as an	
Access	156	IMS BMP Program with DB2 Access	175
Chapter 17. Preparing and Running		Recovery and Restart for IMS BMP	
Generated Programs in MVS Batch	159	Programs	176
Running Main Programs under MVS Batch	159	Chapter 19. Moving Prepared Programs	
Examples of Runtime JCL for MVS Batch		to Other Systems from MVS Systems	177
Programs	160	Moving Prepared Programs To Another	
Running a Main Batch Program with No		MVS System	177
Database Access	160	Maintaining Backup Copies of Production	
Running a Main Batch Program with		Libraries	178
DB2 Access.	161		
Running Main Batch Program with DL/I		<hr/>	
Access	161	Part 4. Utilities for MVS Systems	179
Running a Main Batch Program with			
DB2 and DL/I Access	162	Chapter 20. Using VisualAge Generator	
Recovery and Restart for Batch Programs	164	Server Utilities on MVS Systems	181
		Using the CICS Utilities Menu	181
		New Copy	183
		Diagnostic Message Printing Utility	186

Chapter 21. Diagnostic Control Options	189
Change or View Diagnostic Control Options for a Transaction	190
Change or View Default Diagnostic Control Options	193
Chapter 22. Using the Message File Conversion Utility	195
Converting a Message File to an External Source Format Table.	195
Sample JCL for the Message File Conversion Utility	197
Chapter 23. Using the Parameter Group Utility	199
Chapter 24. IMS Diagnostic Message Print Utility	203

Part 5. Diagnosing Problems on MVS Systems 205

Chapter 25. Diagnosing Problems for VisualAge Generator Server on MVS Systems	207
Detecting Errors	207
Screen Variable Edit Errors	207
File and Database Errors—Category 1	208
File and Database Errors—Category 2	208
File and Database Errors—Category 3	209
Arithmetic Overflow Errors	209
Logic Errors	210
Program Interaction Errors	211
System Errors	213
Reporting Errors	213
Controlling Error Reporting in MVS	
CICS	214
Controlling Error Reporting in IMS Environments	214
Controlling Error Reporting in MVS/TSO and MVS Batch Environments	215
Error Reporting Summary.	215
Transaction Error.	215
Rununit Error	216
Catastrophic error	218
VisualAge Generator Server Error	219
Using the VisualAge Generator Server Error Panel.	220
Printing Diagnostic Information for IMS	221

Printing Diagnostic Information for MVS	
CICS	224
Running the Diagnostic Print Utility	226
Analyzing Errors Detected while	
Running a Program	226
Finding Information in Dumps	227
VisualAge Generator Server ABEND Dumps	227
COBOL or Subsystem ABEND Dumps	228
Information in the VisualAge Generator Server Control Block	229
Information in an Application, Print Services, or Table Profile Block	229
How to Find the Current Position in a Program at Time of Error	230
VisualAge Generator Server Trace Facility	230
Enabling VisualAge Generator Program Source-Level Tracing with a Generation Option	231
Activating a Trace	232
Deactivating a Trace Session	238
Printing Trace Output	239
Reporting Problems for VisualAge Generator Server	240
Problem Reporting Sequence.	242

Chapter 26. Common Messages during Preparation for MVS Systems	245
Common Abend Codes during Preparation	245
MFS Generation Messages	245
DB2 Precompiler and Bind Messages	246
DB2/VSE Precompiler Messages	247
COBOL Compilation Messages	248

Chapter 27. Common System Return Codes for MVS Systems	251
Common EZERT8 Return Codes	251
System Return Code Formats for EZERT8	252
Common System Return Codes in EZERT8	254
VisualAge Generator Server Return Codes	256
Common SQL Return Codes	266
Common DL/I Status Codes	268
Common VSAM Status Codes	269
OPEN request type	269
CLOSE request type.	270
GET/PUT/POINT/ERASE/CHECK/ENDREQ request types	270

COBOL Status Key Values.	271
Chapter 28. VisualAge Generator Server Return Codes and Abend Codes for MVS Systems	273
Return Codes	273
ABEND Codes	273
MVS CICS Environments	274
IMS, IMS BMP, MVS/TSO, and MVS Batch Environments.	276
Chapter 29. Codes from Other Products for MVS Systems	279
Common System Abend Codes for All Environments	279
LE Runtime Messages	281
COBOL Runtime Messages	282
Common COBOL Abend Codes.	282
Common IMS Runtime Messages	283
Common IMS Runtime Abend Codes	284
Common MVS CICS Runtime Messages	285
Common MVS CICS Abend Codes.	286
COBOL Abends under CICS	287
<hr/>	
Part 6. Administering on VSE Systems.	289
Chapter 30. General System Considerations for VSE Systems.	291
Considerations that Affect Performance	291
Generation and Compiler Options	291
Phases in Memory	292
Files and Databases	292
Defining and Loading VSAM Program Data Files	292
Defining VSAM Data Sets.	293
Loading Data in the Files	296
Support for DBCS Terminals	297
Library Space Considerations	298
Extended Addressing Considerations for VisualAge Generator Server	298
Database Considerations	300
Preparing Programs.	300
Checking Access Authorization	301
Moving Program Packages among VSE Systems.	301
Fold Tables.	302
Backing Up Data.	302
Customizing VisualAge Generator Server	302

Chapter 31. System Considerations for VSE CICS	303
Required File Descriptions	303
Segmented and Nonsegmented Processing	304
Using Transient Data Queues for Printing in VSE CICS	305
Processing a DISPLAY Process Option for a Printer Map	305
Specifying Print Destination	306
CICS Terminal Printing.	306
CICS Entries for FZETPRT (DBCS only)	310
Using the New Copy Function	311
Recovery and Database Integrity Considerations	311
Specifying Recovery Options in the CICS Tables	312
Determining When Database Changes Are Committed	312
Considerations that Affect Performance	313
Residency (Phases in Memory) Considerations	313
Work Database Temporary Storage Queue Considerations	315
Conversational and Pseudoconversational Processing	316
Terminal Printing	316
Monitoring and Tuning the System.	317
Using and Allocating Data Files in MVS CICS and VSE CICS.	317
Defining and Loading VSAM Data Files	317
Using Remote Files	319
Defining Transient Data Queues.	320
Considerations for Using DB2/VSE in VSE CICS.	322
Connection Considerations	322
Recovery and Database Integrity Considerations	323
Considerations for Using DL/I in VSE CICS	323
DL/I Database Support	323
Recovery and Database Integrity Considerations	324
Chapter 32. System Considerations for VSE Batch.	325
Using and Allocating VSAM Data Files in VSE Batch	325
Using Sequential Data Files in VSE Batch	325
Determining When Database Changes are Committed.	327

Considerations for Using DB2/VSE in VSE Batch.	328
Connection Considerations	328
Considerations for Processing DB2/VSE Programs	330
Recovery and Database Integrity Considerations	330
Considerations for Using DL/I in VSE Batch.	330
DL/I Support in VSE Batch Programs	330
Recovery and Database Integrity Considerations	331
Performance Considerations for VSE Batch	331
Runtime JCL	331

Part 7. Preparing and Running Generated Applications on VSE Systems. 333

Chapter 33. Output of Program Generation on VSE Systems 335

List of Program Preparation Steps after Program Generation.	335
Outputs of Generation	335
Objects Generated for Programs.	338
Objects Generated for Tables	339
Objects Generated for Map Groups.	339
Objects Generated for All Part Types (Programs, Tables, and Map Groups)	340

Chapter 34. Preparing Generated Programs on VSE Systems 343

Allocating Preparation Data Sets	343
Running the Preparation Job	343
Error Processing for Preparation Jobs	343
Analyzing the Preparation Listing	344
Analyzing the DB2/VSE Precompile Step	344
Analyzing the CICS Translator Step	345
Analyzing the Compile Step	345
Analyzing the Link-edit Step	345
Preparing to Run a DB2/VSE Program	346
Granting Authority to Program Users	346
Defining Synonyms for DB2/VSE Tables	347
Defining a Program Specification Block	347

Chapter 35. Preparing and Running Generated Programs in VSE CICS 349

Modifying CICS Resource Tables	349
Program Entries (PPT)	349

Transaction Entries (PCT)	351
Destination Control Table Entries (DCT)	354
File Control Table Entries (FCT)	354
Using Remote Programs, Transactions, or Files	354
Using VSE/POWER Spool Files	355
Modifying CICS Startup JCL	355
Making New Programs Available in the VSE CICS Environment	355
Making Programs Resident	356
Running Programs under VSE CICS	356
Starting the Transaction in VSE CICS	356
Controlling Diagnostic Information in the CICS Environment	356

Chapter 36. Preparing and Running Generated Programs in VSE Batch 357

Making VisualAge Generator Server, COBOL, and the Generated Programs Available	357
Creating or Modifying Runtime JCL	357
Tailoring JCL before Generation	358
Modifying Runtime JCL	359
Running Main Programs under VSE Batch	361
Examples of Runtime JCL for VSE Batch Programs	362
Running a Main Batch Program with No Database Access	363
Running a Main Batch Program with DB2/VSE Access	363
Running a Main Batch Program with DL/I Access	365
Running a Main Batch Program with DB2/VSE and DL/I Access	367
Recovery and Restart for Batch Programs	369

Chapter 37. Moving Prepared Programs to Other Systems from VSE Systems 371

Moving Prepared Programs to Another VSE System	371
Maintaining Backup Copies of Production Libraries	371

Part 8. Utilities for VSE Systems 373

Chapter 38. Using VisualAge Generator Server Utilities on VSE Systems 375

Using the CICS Utilities Menu	375
New Copy	377

Diagnostic Message Printing Utility . . .	380
Chapter 39. Diagnostic Control Options	383
Change or View Diagnostic Control Options for a Transaction	384
Change or View Default Diagnostic Control Options	387
Chapter 40. Using the Message File Conversion Utility	389
Converting a Message File to an External Source Format Table.	389
Sample JCL for the Message File Conversion Utility	391
Chapter 41. Using the Parameter Group Utility	393

Part 9. Diagnosing Problems on VSE Systems 397

Chapter 42. Diagnosing Problems for VisualAge Generator Server on VSE Systems	399
Detecting Errors	399
Reporting Errors	405
Controlling Error Reporting in VSE CICS	405
Controlling Error Reporting in VSE Batch Environments	405
Error Reporting Summary.	406
Using the VisualAge Generator Server Error Panel.	408
Printing Diagnostic Information for VSE CICS.	408
Running the Diagnostic Print Utility	410
Analyzing Errors Detected while Running a Program	410
Finding Information In Dumps	411
VisualAge Generator Server ABEND Dumps	411
COBOL or Subsystem ABEND Dumps	412
Information in the VisualAge Generator Server Control Block	413
Information in a Program, Print Services, or Table Profile Block	413
How to Find the Current Position in a Program at Time of Error	414
Using the VisualAge Generator Server Trace Facility	414

Enabling VisualAge Generator Program Source-Level Tracing with a Generation Option	415
Activating a Trace	415
Deactivating a Trace Session	422
Printing the Trace Output	422
Reporting Problems for VisualAge Generator Server	423
Problem Reporting Sequence	425

Chapter 43. Common Messages during Preparation for VSE Systems	427
VSE System and Job Control Messages	427
Workstation File Transfer Messages	427
DB2/VSE Precompiler Messages	428
COBOL Compilation Messages	429

Chapter 44. Common System Return Codes for VSE Systems	431
Common EZERT8 Return Codes	431
System Return Code Formats for EZERT8.	432
Common System Return Codes in EZERT8.	433
VisualAge Generator Server Return Codes	435
Common DB2/VSE Return Codes	445
Common DL/I Status Codes	447
Common VSAM Status Codes	448
OPEN request type	448
CLOSE request type.	448
GET/PUT/POINT/ERASE/CHECK/ENDREQ request types	449
COBOL Status Key Values.	449

Chapter 45. VisualAge Generator Server Return Codes and Abend Codes for VSE Systems	451
Return Codes	451
ABEND Codes	451
VSE CICS and VSE Batch Environments	451
VSE Batch Environment	454

Chapter 46. Codes from Other Products for VSE Systems	457
Common System Abend Codes for All Environments	457
LE Runtime Messages	458
Common COBOL Abend Codes.	459

Common VSE CICS and VSE Batch Runtime Messages	459
Common VSE CICS and VSE Batch Abend Codes	461
COBOL Abends under CICS	462
Common Runtime Messages for DL/I DOS/VS	462
Common Runtime Messages for SQL/DS	463

Part 10. Administering on VM Systems. 465

Chapter 47. General System Considerations for VM Systems	467
Virtual Machine Size	467
Considerations that Affect Performance	467
Generation and Compiler Options	467
Files and Databases	468
Defining and Managing VSAM Program Data Files	468
Defining Data Files	469
Loading Data into a VSAM File	470
Support for DBCS Terminals	471
Extended Addressing Considerations for VisualAge Generator Server	471
Database Considerations	472
Preparing Programs	472
Checking Access Authorization	473
Moving Program Packages Among VM Systems	473
Fold Tables	474
Backing Up Data	474
Customizing VisualAge Generator Server	474
Chapter 48. System Considerations for VM CMS and VM batch	475
Required and Optional File Descriptions	475
Printing Considerations for VM CMS and VM batch	476
Making VisualAge Generator Server, COBOL, and the Generated Modules Available	477
Recovery Considerations	477
Using VSAM Program Data Files in VM	478
Considerations for Using DB2/VSE in VM	479
Connection Considerations for SQL Programs	479
Starting SQL Programs in VM	481

Recovery and Database Integrity Considerations	482
---	-----

Part 11. Preparing and Running Generated Applications on VM Systems. 483

Chapter 49. Output of Program Generation on VM Systems.	485
List of Program Preparation Steps after Program Generation.	485
Outputs of Generation	485
Objects Generated for Programs.	487
Objects Generated for Tables	488
Objects Generated for Map Groups.	488
Objects Generated for All Part Types	489
Chapter 50. Preparing Generated Programs on VM Systems	491
Running the Preparation REXX Exec	491
Analyzing the Preparation Listing	491
Analyzing the SQL Precompile Step	491
Analyzing the Compile Step	491
Analyzing the Link-Edit Step.	492
Preparing to Run an SQL Program	493
Granting Authority to Program Users	493
Defining Synonyms for SQL Tables.	493
Using LE Runtime Options for VM CMS and VM batch.	494
Chapter 51. Modifying the Sample Runtime REXX Execs.	497
Tailoring Sample REXX Exec Templates Before Generation	498
Modifying the Sample Runtime REXX Execs After Generation	500
File Allocation Considerations	500
Chapter 52. Running a Generated Program on VM Systems	511
Making VisualAge Generator Server for MVS, VSE, and VM, COBOL, and the Generated Modules Available	511
Running Main Programs under VM	512
VM Saved Shared Segments	512
Chapter 53. Moving Prepared Programs to Other Systems from VM Systems	515

Moving Prepared Programs To Another VM System	515
Maintaining Backup Copies of Production Libraries	515
Chapter 54. Execs for Preparing and Running Programs on VM Systems . . .	517
ELACALLC EXEC	518
ELAPREP EXEC	521
ELARUN EXEC	527
ELASQLAA EXEC	528

Part 12. Utilities for VM Systems 531

Chapter 55. Using the Message File Conversion Utility on VM Systems . . .	533
Converting a Message File to an External Source Format Table.	533
Sample Exec for the Message File Conversion Utility	535

Part 13. Diagnosing Problems on VM Systems 539

Chapter 56. Diagnosing Problems for VisualAge Generator Server on VM Systems	541
Detecting Errors	541
Reporting Errors	546
Controlling Error Reporting in VM CMS and VM Batch Environments.	546
Error Reporting Summary.	546
Using the VisualAge Generator Server Error Panel.	547
Analyzing Errors Detected while Running a Program	548
Finding Information in Dumps	549
VisualAge Generator Server ABEND Dumps	549
COBOL or Subsystem ABEND Dumps Information in the VisualAge Generator Server Control Block	551
Information in a Program, Print Services, or Table Profile Block	551
How to Find the Current Position in a Program at Time of Error	552
VisualAge Generator Server Trace Facility	552

Enabling VisualAge Generator Program Source-Level Tracing with a Generation Option	553
Activating a Trace	553
Deactivating a Trace Session	557
Printing Trace Output	557
Reporting Problems for VisualAge Generator Server	557
Problem Reporting Sequence	559

Chapter 57. Common Messages during Preparation for VM Systems	561
COBOL Compilation Messages	561
DB2/VSE Precompiler Messages	562

Chapter 58. Common System Return Codes for VM Systems	565
Common EZERT8 Return Codes	565
System Return Code Formats for EZERT8.	565
Common System Return Codes in EZERT8.	566
VisualAge Generator Server Return Codes	567
Common SQL Return Codes	573
Common VSAM Status Codes	575
OPEN request type	575
CLOSE request type.	576
GET/PUT/POINT/ERASE/CHECK/ENDREQ request types	576
COBOL Status Key Values.	576

Chapter 59. VisualAge Generator Server Return Codes and Abend Codes for VM Systems	579
Return Codes	579
ABEND Codes	579

Chapter 60. Codes from Other Products for VM Systems	581
Common System Abend Codes for All Environments	581
COBOL Runtime Messages	582
COBOL/370 Runtime Messages.	583
Common COBOL Abend Codes.	584

Part 14. Messages for MVS, VSE, and VM Systems 585

Chapter 61. VisualAge Generator Server Messages for MVS, VSE, and VM	
Systems	587
Languages Available	587
VisualAge Generator Server Message	
Format	587
Reporting Problems to IBM	589
Messages	590

Appendix. Understanding Syntax	
Diagrams	649
Index	651
Readers' Comments — We'd Like to Hear from You	671

Notices

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

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood NY 10594, U.S.A.

Licenseses 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 the SWS General Legal Counsel, IBM Corporation, Department TL3 Building 062, P. O. Box 12195, Research Triangle Park, NC 27709-2195. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

IBM has made reasonable efforts to ensure the accuracy of the information contained in this publication. If a softcopy of this publication is provided to you with the product, you should consider the information contained in the softcopy version the most recent and most accurate. However, this publication is presented "as is" and IBM makes no warranties of any kind with respect to the contents hereof, the products listed herein, or the completeness or accuracy of this publication.

IBM may change this publication, the product described herein, or both.

Trademarks

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

AD/Cycle
AIX
AIX/6000
CICS
CICS OS/2
CICS/ESA
CICS/MVS
CICS/VSE
CICS/6000
COBOL/370
COBOL/400
DATABASE 2
DataJoiner
DB2
DB2/2
DB2/400
DB2/6000
Distributed Database Connection Services/2
IBM
IMS
IMS/ESA
Language Environment
Operating System/2
OS/2
OS/400
RS/6000
SAA
SQL/DS
SQL/400
VisualAge
VisualGen

Virtual Machine/Enterprise Systems Architecture
VM/ESA

The following terms are trademarks of other companies:

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Terminology Used in This Document

Unless otherwise noted in this publication, the following references apply:

- MVS CICS applies to Customer Information Control System/Enterprise Systems Architecture (CICS/ESA) systems.
- CICS applies to CICS/VSE, CICS/ESA, CICS Operating System/2 (CICS OS/2), CICS for AIX, and CICS for Windows NT.
- CICS for Windows NT refers to IBM Transaction Server for Windows NT Version 4.0.
- IMS/VS applies to Information Management System/Enterprise System Architecture (IMS/ESA) and IMS/ESA Transaction Manager systems.
- IMS applies to IMS/ESA and IMS/ESA Transaction Manager, and to message processing program (MPP), IMS Fast Path (IFP), and batch message processing (BMP) regions. IMS/VS is used to distinguish MPP and IFP regions from the IMS BMP target environment.
- LE applies to the IBM Language Environment for MVS and VM.
- COBOL applies to any of the following types of COBOL:
 - IBM VisualAge for COBOL for OS/2
 - ILE COBOL/400
 - IBM COBOL for VSE
 - IBM COBOL for MVS and VM
- “Region” or “CICS region” correspond to the following:
 - MVS CICS region
 - IMS region
 - VSE CICS partition
 - CICS OS/2 system
 - CICS for AIX system
 - CICS for Windows NT system
- DB2/VSE refers to SQL/DS Version 3 Release 4 or later. Any references to SQL/DS refer to DB2/VSE and SQL/DS on VM. In addition, any references to SQL/400 refer to DB2/400.
- OS/2 CICS applies to CICS Operating System/2 (CICS OS/2).

- Workstation applies to a personal computer, not an AIX workstation.
- The make process applies to the generic process not to specific make commands, such as make, nmake, pmake, polymake.
- Unless otherwise noted, references to VM apply to Virtual Machine/Enterprise Systems Architecture (VM/ESA) environments.
- References to VM batch apply to any batch facility running on VM.
- Windows applies to Windows Version 3.11, Windows 95, and Windows NT.

About This Document

This manual provides information about customizing and administering VisualAge Generator Server for MVS, VSE, and VM in the following environments:

- MVS
- MVS CICS
- MVS/TSO
- MVS batch
- IMS
- VM CMS
- VM Batch
- VSE
- VSE CICS
- VSE Batch

It also provides information to enable you to prepare VisualAge Generator programs for running in the MVS, VSE, and VM environments.

Note: Hereafter in this book, VisualAge Generator Server for MVS, VSE, and VM is referred to simply as “VisualAge Generator Server.”

Who Should Use This Document

This manual is intended for system administrators and system programmers responsible for installing, maintaining, and administering VisualAge Generator Server. It provides information to complete the following tasks:

- Manage system requirements
- Set up requirements for system printing
- Manage file utilization and conflicts
- Set up and use batch processing

This manual is also intended for use by the programmers responsible for preparing and running VisualAge Generator programs. It provides information on the following items:

- Outputs of the generation process
- How to prepare generated programs for running
- Error codes

- How to use VisualAge Generator Server utilities
- How to diagnose and report problems

Documentation provided with VisualAge Generator

The documents are provided in one or more of the following formats:

- Printed and separately ordered using the individual form number.
- Printed and ordered as a set using the bill of forms number or the document kit number.
- Online book files (.pdf) on the product CD-ROM. Adobe Acrobat Reader is used to view the manuals online and to print desired pages. Adobe Acrobat Reader is included with VisualAge Generator.

The following documents are shipped with the VisualAge Generator CD and can be ordered separately using the individual order numbers:

- *VisualAge Generator Getting Started* (GH23-6601-02) ¹
- *VisualAge Generator Installation Guide* (GH23-6598-02) ¹
- *VisualAge Generator Programmer's Guide to Building Parts for Fun and Profit* (SH23-6608-02)
- *Introducing VisualAge Generator Templates* (GC34-4728-01)

The following documents can be ordered as a set using the bill of forms number (SBOF-6727) or the document kit number, or separately using the individual order numbers.

- *VisualAge Generator Design Guide* (SH23-6607-01) ¹
- *VisualAge Generator Client/Server Communications Guide* (SH23-6602-02) ¹
- *VisualAge Generator Generation Guide* (SH23-6606-02) ¹
- *VisualAge Generator Messages and Problem Determination Guide* (GH23-6597-02) ¹
- *VisualAge Generator Server Guide for OS/2, AIX, Windows NT, and HP-UX* (SH23-6603-01) ¹
- *VisualAge Generator Programmer's Reference* (SH23-6605-02) ¹
- *VisualAge Generator System Development Guide* (SG24-4230-02) ¹
- *VisualAge Generator Templates User's Guide - Standard Functions* (SC34-4729-01) ¹

The following hardcopy documents are available in printed form for VisualGen Host Services for OS/400 and VisualAge Generator Server for MVS, VSE, and VM:

1. These documents are available as softcopy PDF files on the product CD.

- *Running VisualGen Applications on OS/400* (SH23-6549-01)
- *VisualAge Generator Server Guide for MVS, VSE, and VM* (SH23-0256-00)

The following information is also available for VisualAge Generator:

- *VisualAge Generator External Source Format Reference* (SH23-6609-01)
- *Migrating Cross System Product Applications to VisualAge Generator* (SH23-0244-00)
- *VisualAge Generator Guide to Migrating MSLs to ENVY* (SH23-0252-01) ¹
- *VisualAge Generator Templates Reference* (SC34-4730-01) ¹

Related information

Several sources of information on the VisualAge products are available.

Product information

The following publications are provided with the VisualAge for Smalltalk products. They can help you find information about the products. The *VisualAge Resource Catalog* lists additional books on related subjects such as object-oriented programming and user interface design.

Documentation provided with VisualAge for Smalltalk:

- *VisualAge for Smalltalk Getting Started* (SC34-4535)
- *VisualAge for Smalltalk User's Guide* (SC34-4518)
- *VisualAge for Smalltalk User's Reference*
- *IBM Smalltalk Programmer's Reference* (SC34-4493)
- *IBM Smalltalk User's Guide* (SC34-4536)
- *VisualAge for Smalltalk Image Component Developer's Guide and Reference*
- *VisualAge for Smalltalk AS/400 Connection Guide and Reference*
- *VisualAge for Smalltalk Communications/Transactions Guide and Reference*
- *VisualAge for Smalltalk Distributed User's Guide*
- *VisualAge for Smalltalk Web Connection User's Guide*
- *VisualAge for Smalltalk Features Class Guide*
- *VisualAge for Smalltalk CICS & IMS Connection User's Guide and Reference*
- *VisualAge for Smalltalk Connection for Lotus Notes Guide and Reference*
- *VisualAge for Smalltalk Installation Guide*
- *VisualAge for Smalltalk Multimedia Guide and Reference*
- *VisualAge for Smalltalk Programmer's Guide to Building Parts for Fun and Profit* (SC34-4496)
- *VisualAge for Smalltalk Reports Guide and Reference*

- *VisualAge for Smalltalk Server Enablement for MVS/ESA User's Guide (SC34-4615)*
- *VisualAge for Smalltalk Server for MVS/ESA User's Guide and Reference (SC34-4618)*
- *VisualAge for Smalltalk Tivoli Connection Guide and Reference*
- *VisualAge for Smalltalk Visualization Tools User's Guide*

Other related publications

- *VisualAge Resource Catalog (GC09-2455)*
- *VisualAge: Building GUIs for Existing Applications (GG24-4244)*
- *VisualAge: Concepts and Features (GG24-3946)*
- *Smalltalk With Style (SR23-7337)*
- *IBM Smalltalk: The Language (SR28-5628)*
- *IBM Smalltalk Programming for Windows and OS/2 (SR23-7346)*

Part 1. Preparing to Install on MVS, VSE and VM Systems

Chapter 1. Preparing for the Installation of VisualAge Generator Server for MVS, VSE, and VM	3	Making VisualAge Generator Server for MVS, VSE, and VM, COBOL, and the Generated Modules Available	25
Chapter 2. Hardware Requirements for VisualAge Generator Server for MVS, VSE, and VM	5	MVS CICS Installation Considerations.	26
Chapter 3. Software Requirements for VisualAge Generator Server for MVS, VSE, and VM	7	CICS/ESA Considerations.	26
VisualAge Generator Server for MVS	7	DL/I Considerations	26
VisualAge Generator Server for VSE	9	DB2 Considerations	26
VisualAge Generator Server for VM	10	Security Considerations	27
Chapter 4. Storage Requirements for VisualAge Generator Server for MVS, VSE, and VM	13	Monitoring and Tuning	27
Virtual Storage Requirements	13	MVS CICS Utilities	27
VisualAge Generator Server for MVS, VSE, and VM Load Module Storage	13	Client/Server Processing Considerations	27
Application Load Module Storage	14	Using the /DATA Generation Option	28
COBOL Dynamic Storage	15	Modifying MVS CICS Resource Tables	28
VisualAge Generator Server for MVS, VSE, and VM Dynamic Storage.	16	Using Spool Files.	29
Storage Requirements for MVS CICS and VSE CICS	17	Terminal Considerations	29
Disk Storage Requirements for VisualAge Generator Server for MVS, VSE, and VM	18	Temporary Storage	29
Work Database Space For Segmented Applications	18	IMS Installation Considerations	29
Chapter 5. Installation Considerations for MVS, VSE, and VM Systems	21	IMS/ESA Exploitation	29
Installation and Language-Dependent Options for MVS, VSE, and VM.	21	DB2 Considerations	30
MVS Batch and MVS/TSO Installation Considerations	24	Security Considerations	30
DL/I Considerations	24	Monitoring and Tuning	30
DB2 Considerations	25	IMS System Definition	31
Security Considerations	25	IMS Control Region.	31
		Work Database	31
		DL/I Work Database Considerations	32
		DB2 Work Database Considerations	32
		VSE Batch Installation Considerations.	32
		DL/I Considerations	32
		DB2/VSE Considerations	33
		Making VisualAge Generator Server for MVS, VSE, and VM, COBOL, and the Generated Modules Available	33
		VSE CICS Installation Considerations	33
		DL/I Considerations	34
		DB2/VSE Considerations	34
		Security Considerations	34
		Monitoring and Tuning	35
		VSE CICS Utilities	35
		Client/Server Processing Considerations	35
		Using the /DATA=24 Generation Option	35
		Modifying VSE CICS Resource Tables	36
		Using Spool Files.	36
		Terminal Considerations	36

Temporary Storage	36	Security Considerations	39
VM Batch and VM CMS Installation		Performance Considerations	40
Considerations	37	Customizing JCL Procedures	41
SQL/DS Considerations	37	Modifying the COBOL Working Storage	
Security Considerations	37	Initialization Option.	41
Chapter 6. Customizing VisualAge		Using Generated Programs with PL/I	
Generator Server for MVS, VSE, and VM	39	Programs	41
General Customization Considerations for			
MVS, VSE, and VM Systems	39		

Chapter 1. Preparing for the Installation of VisualAge Generator Server for MVS, VSE, and VM

The following chapters describe the requirements for installing VisualAge Generator Server for MVS, VSE, and VM.

Installation of the initial code shipment and all updates is the responsibility of the customer.

After selecting the production environments, do the following to prepare for the installation of VisualAge Generator Server for MVS, VSE, and VM:

- Determine the hardware, software, and storage requirements for the production environments selected.
- Install the hardware and software required by VisualAge Generator Server for MVS, VSE, and VM. Refer to the *Program Directory for VisualAge Generator Server* for your MVS, VSE, or VM environment for more information about installation.
- Collect information before customization.
- Understand specific environment considerations before defining applications.

Before installing VisualAge Generator, check the preventive service planning (PSP) facility buckets for recommended maintenance to be applied after the installation is complete. The PSP facility contains the latest information that is available for VisualAge Generator. To access the PSP buckets for the prerequisite maintenance information, use the electronic link with IBM service (for example, IBMLink, INFO/ACCESS, or SERVICELINK) or contact the IBM Support Center.

For countries where IBM Talklink and CompuServe forums are available, recommended maintenance information can be obtained. For Talklink users, use the VISUALGN forum to gain access to the maintenance information. For CompuServe users, from any ! prompt, type GO VISUAL, then select VAGEN.

In addition, refer to the PSP section of the program directory for preventive service information on the environment you are installing.

Chapter 2. Hardware Requirements for VisualAge Generator Server for MVS, VSE, and VM

Table 1 shows the **minimum** hardware requirements for VisualAge Generator Server for MVS, VSE, and VM.

Table 1. Hardware Requirements for VisualAge Generator Server for MVS, VSE, and VM

Processor	Any IBM System/370 or IBM System/390 processor that supports any operating system listed in “Chapter 3. Software Requirements for VisualAge Generator Server for MVS, VSE, and VM” on page 7.
Storage devices	<ul style="list-style-type: none">• For MVS and VSE:<ul style="list-style-type: none">– Any direct-access storage device supported by the Virtual Storage Access Method (VSAM) and the operating system– For system libraries, any devices supported by the operating system are allowed.• For VM:<ul style="list-style-type: none">– Any CMS Minidisk or Shared File System directory
Tape drive	A magnetic tape drive supported by the operating system (required for installation and maintenance)
Printer	Any printer that supports your applications that perform printing.
Terminals	At least one of each of these: <ul style="list-style-type: none">• Operator terminal• Display terminal
Library devices	For system libraries, any devices supported by the operating system.

Chapter 3. Software Requirements for VisualAge Generator Server for MVS, VSE, and VM

The following sections describe the software requirements for VisualAge Generator Server for MVS, VSE, and VM.

VisualAge Generator Server for MVS

Table 2 shows the **minimum** software requirements for VisualAge Generator Server for MVS.

Table 2. Software Requirements for VisualAge Generator Server for MVS

MVS Requirements	
MVS system	One of the following MVS environments, including MVS batch functions with the appropriate prerequisites: <ul style="list-style-type: none">• MVS/ESA Version 4.2 or later (Program number 5695-047 or 5695-048)• OS/390 Version 1.1 or later (Program number 5645-001) Notes: <ol style="list-style-type: none">1. MVS/ESA Version 5.1 or later is required for using DCE communications.2. OS/390 Version 1.3 or later is required for OpenEdition web server support.
SMP/E	System Modification Program/Extended (SMP/E) Version 1.8.1 or later (Program number 5668-949)
ISPF products	Interactive System Productivity Facility (ISPF) Version 4.1 (Program number 5655-042)
Assembler products	IBM High Level Assembler Version 1.2 or later (Program number 5696-234)
Compile products	One of the following COBOL products for compiling generated COBOL programs: <ul style="list-style-type: none">• IBM COBOL for MVS and VM Version 1.2 (Program number 5688-197)• IBM COBOL for OS/390 and VM Version 2.1 or later (Program number 5648-A25)
Runtime products	IBM Language Environment for MVS & VM Version 1.5 or later (Program number 5688-198)

Table 2. Software Requirements for VisualAge Generator Server for MVS (continued)

File Transfer	For transferring files between MVS and a workstation, one of the following: <ul style="list-style-type: none"> • TCP/IP Version 3.1 or later (Program number 5665-HAL) • IBM 3270/PC File Transfer Program/MVS Release 1 (Program number 5665-311)
----------------------	--

Table 3 shows the **optional** software requirements for VisualAge Generator Server for MVS.

Table 3. Optional Software Requirements for VisualAge Generator Server for MVS

MVS CICS Software	CICS/ESA Version 3.3 or later (Program number 5685-083) Note: For OS/390 Open Edition web server support, CICS/ESA Version 4.1 or later (Program number 5655-018) is required.
IMS products	To use hierarchical databases or run IMS programs, one of the following IMS products: <ul style="list-style-type: none"> • IMS/ESA Database Manager Version 4.1 or later (Program number 5685-012). The IMS/ESA Transaction Manager Version 4.1 or later (Program number 5685-013) is required in addition to the IMS/ESA Database Manager Version 4.1 or later for MPPs, Fast Path regions, or Fast Path main-storage databases.
Relational Databases	IBM DATABASE 2 (DB2) Version 3.1 or later (Program number 5685-DB2) Note: IBM DB2 for MVS/ESA Version 4 or later is required for participation of a DB2 database in a DRDA distributed unit of work.
Communications support	For DCE client/server communications, IBM OpenEdition DCE Application Support for MVS/ESA Version 1.2 (program number 5655-064)

Table 3. Optional Software Requirements for VisualAge Generator Server for MVS (continued)

OS/390 OpenEdition web server support	<p>For a web server that accesses VisualAge Generator server programs, the following are required:</p> <ul style="list-style-type: none"> • OS/390 Version 1.3 or later (Product number 5645-001) with MVS OpenEdition active and TCP/IP enabled • CICS/ESA Version 4.1 or later (Product number 5655-018) • Java Development Kit 1.1.4 for OS/390 or later (Program number 5655-A46) • A web server application, such as one of the following, must be installed on the MVS OpenEdition platform: <ul style="list-style-type: none"> – IBM Internet Connection Secure Server (ICSS) for OS/390 Version 2.1 or later (Product number 5697-B14) – Domino Go Webserver for OS/390 Version 4.6.1 (Product number 5697-C58) <p>Note: If Domino Go Webserver is used, OS/390 Version 2.4 (Product number 5647-A01) is required.</p>
Interactive Chart Utility access	<ul style="list-style-type: none"> • To access the Interactive Chart Utility from a VisualAge Generator program, Graphical Data Display Manager/MVS (GDDM/MVS) Version 3.1.1 or later (Program number 5695-167) is required.

VisualAge Generator Server for VSE

Table 4 shows the **minimum** software requirements for VisualAge Generator Server for VSE.

Table 4. Software Requirements for VisualAge Generator Server for VSE

VSE Requirements	
VSE System	<p>One of the following:</p> <ul style="list-style-type: none"> • VSE/ESA Version 1.3.2 or later (Program number 5750-ACD) • VSE/ESA Version 2.1 or later (Program number 5790-VSE)
CICS	<p>IBM CICS/VSE Version 2.3 or later (Program number 5686-026)</p> <p>Note: To use the full spool file function (including trace functions) on VSE CICS, you must have the CICS/VSE Report Controller feature on your system.</p>
Assembler	<p>IBM High Level Assembler Version 1.2 or later (Program number 5696-234)</p>

Table 4. Software Requirements for VisualAge Generator Server for VSE (continued)

COBOL products for generation	Both of the following COBOL products for generating COBOL programs: <ul style="list-style-type: none"> • Language Environment for VSE Version 1.4 or later (Program number 5686-094) • IBM COBOL for VSE Version 1.1 or later (Program number 5686-068)
COBOL products for runtime	For running generated COBOL programs: <ul style="list-style-type: none"> • Language Environment for VSE Version 1.4 or later (Program number 5686-094)

Table 5 shows the **optional** software requirements for VisualAge Generator Server for VSE.

Table 5. Optional Software Requirements for VisualAge Generator Server for VSE

Hierarchical database access	DL/I DOS/VS Version 1.10 or later (Program number 5746-XX1) at a level that is compatible with IBM CICS/VSE Version 2.3 (Program number 5686-026)
Relational database access	One of the following: <ul style="list-style-type: none"> • Structured Query Language/Data System (SQL/DS) Version 3.5 or later (Program number 5688-103) • DB2 Server for VSE and VM Version 5.1 or later (Program number 5648-158)
Communications support	For client/server processing between VisualAge Generator client/server application systems running under CICS OS/2 and CICS/VSE, IBM CICS/VSE Version 2.3 or later (Program number 5686-026)
Interactive Chart Utility access	For accessing the Interactive Chart Utility from a VisualAge Generator program, Graphical Data Display Manager/VSE (GDDM/VSE) Version 3.1 or later (Program number 5686-057) at a level that is compatible with IBM CICS/VSE Version 2.3 (Program number 5686-026)

VisualAge Generator Server for VM

Table 6 shows the **minimum** software requirements for VisualAge Generator Server for VM.

Table 6. Software Requirements for VisualAge Generator Server for VM

VM Requirements

Table 6. Software Requirements for VisualAge Generator Server for VM (continued)

VM system	One of the following: <ul style="list-style-type: none"> • VM/ESA Version 1.2.2 (Program number 5684-112, with RSU9403 service level and APARs VM60613 and VM60578) • VM/ESA Version 2.1 or later (Program number 5654-030), with RSU9403 service level and APARs VM60613, VM60578, and PN83993
Assembler	IBM High Level Assembler/MVS and VM and VSE Version 1.2 or later (Program number 5696-234)
COBOL products for runtime	<ul style="list-style-type: none"> • IBM Language Environment for MVS and VM Version 1.5 or later (Program number 5688-198) • One of the following: <ul style="list-style-type: none"> – IBM COBOL for MVS and VM Version 1.2 or later (Program number 5688-197) – IBM COBOL for OS/390 and VM Version 2.1 or later (Program number 5648-A25)
File Transfer	One of the following: <ul style="list-style-type: none"> • IBM TCP/IP for VM Version 2.3 or later (Program number 5735-FAL) • IBM 3270/PC File Transfer Program for VM (Program number 5664-281)

Table 7 shows the **optional** software requirements for VisualAge Generator Server for VM.

Table 7. Optional Software Requirements for VisualAge Generator Server for VM

Relational database access	To use relational databases, one of the following: <ul style="list-style-type: none"> • Structured Query Language/Data System (SQL/DS) Version 3.5 or later (Program number 5688-103) • IBM DB2 Server for VSE and VM Version 5.1 or later (Program number 5648-158)
Communications	OpenEdition Distributed Computing Environment for VM/ESA Version 2.1 installed
Interactive Chart Utility access	For accessing the Interactive Chart Utility from a VisualAge Generator program: <ul style="list-style-type: none"> • Graphical Data Display Manager/VM (GDDM/VM) Version 3.1.0 with APAR PN39848 (Program number 5684-168-01) • Graphical Data Display Manager/PGF Version 2.1.1 or later (Program number 5668-812-01)

Chapter 4. Storage Requirements for VisualAge Generator Server for MVS, VSE, and VM

The following sections give approximate estimates of VisualAge Generator Server for MVS, VSE, and VM storage use by type of storage.

Virtual Storage Requirements

An application requires virtual storage for the following:

- VisualAge Generator Server for MVS, VSE, and VM load modules
- Application load modules
- COBOL dynamic area
- VisualAge Generator Server for MVS, VSE, and VM dynamic area

CICS applications also use specialized CICS storage facilities.

VisualAge Generator Server for MVS, VSE, and VM Load Module Storage

Most of the modules in the runtime function are not linked with the generated applications. Only one copy of these modules needs to be available for use by all applications generated with VisualAge Generator Server for MVS, VSE, and VM.

For MVS and VSE, these modules can be in a library (STEPLIB or LIBDEF), placed in the link pack area (LPA) for MVS, in a library managed by the Virtual Lookaside Facility (VLF) and Library Lookaside (LLA) features of MVS/ESA, or placed in the shared virtual area (SVA) list for VSE. For IMS, you might want to preload the modules. For CICS, you might want to make the modules resident. Refer to the VisualAge Generator Server for MVS, VSE, and VM program directory for a list of LPA and SVA eligible load modules.

For VM, these modules can be in a library (LOADLIB) on a CMS Minidisk or in a Shared File System directory. Table 8 shows the reentrant load module storage estimates.

Table 8. VisualAge Generator Server for MVS, VSE, and VM Reentrant Load Module Storage Estimates

Function	Size	RMODE
MVS CICS and VSE CICS base services	240KB	ANY

Table 8. VisualAge Generator Server for MVS, VSE, and VM Reentrant Load Module Storage Estimates (continued)

Function	Size	RMODE
MVS CICS and VSE CICS base services, 24-bit addressing mode	8KB	24
IMS, IMS BMP, MVS/TSO, MVS batch, VSE batch, VM CMS, and VM batch base services	255KB	ANY
IMS, IMS BMP, MVS/TSO, MVS batch, VSE batch, VM CMS, VM batch base services, 24-bit addressing mode	10KB	24
Double-byte language ASCII/EBCDIC code conversion tables	Chinese - 50KB	ANY

Application Load Module Storage

Application load module storage is the storage required for generated COBOL applications. The load modules are created by link-editing the generated COBOL applications produced by VisualAge Generator generation facility. The size of the load module can be determined from the linkage editor module map. The size varies depending on the functions utilized with the applications.

The application load module storage includes all application programs, map group format modules, print services programs, and table programs used by a batch job step, transaction, VM or MVS/TSO invocation. It also includes the small VisualAge Generator Server for MVS, VSE, and VM database manager stub programs that are statically linked with the programs. The load modules produced by link-editing the generated programs are reentrant. Each module can be linked with RMODE(ANY) so that the load module can reside in extended storage.

The size of the VisualAge Generator Server for MVS, VSE, and VM modules that are linked with each application program, print services program, or table program is shown in Table 9. These estimates should be added to the application load module size to determine the overall load module size.

Table 9. VisualAge Generator Server for MVS, VSE, and VM Statically Linked Module Storage Estimates

Environment	Application	Print service program	Table program
MVS CICS and VSE CICS	2.5KB	1KB	1KB
IMS	1KB	1KB	1KB

Table 9. VisualAge Generator Server for MVS, VSE, and VM Statically Linked Module Storage Estimates (continued)

Environment	Application	Print service program	Table program
MVS/TSO, MVS batch, VSE batch, VM CMS, VM batch, IMS BMP	1.3KB	1KB	1KB
Note: VisualAge Generator Server for MVS, VSE, and VM modules are not statically linked with a map group format module.			

COBOL Dynamic Storage

Application load modules acquire dynamic storage while they are running. The COBOL runtime library requires this storage for application data structures such as records, maps, tables, and working storage. The storage includes both the internal and external data structures.

The COBOL /DATA generation option determines whether to acquire storage below the 16MB line. The procedures shipped with VisualAge Generator Server for MVS, VSE, and VM enable the /DATA generation option to control the value for the COBOL DATA compiler option. /DATA=24 is the default generation option for the VM environment. /DATA=24 is also the default generation option for the non-CICS environments, but must be specified for IMS/VS systems, and for MVS/TSO and MVS batch DL/I applications that use IMS/VS, for DL/I support. The /DATA=24 generation option must be specified for DL/I applications running in a VSE batch environment. /DATA=24 is also required if an application calls another application or program that is linked as AMODE(24). Tables and print services programs must also use /DATA=24 if any application being used is linked AMODE(24).

The /DATA=31 generation option can be used when IMS/ESA is used for DL/I support, or when MVS/TSO and MVS batch applications do not use DL/I. /DATA=31 is required for MVS CICS and VSE CICS applications unless the dynamic storage requirement for the program is less than 64KB.

The amount of storage required for internal data structures is listed in the compile listing of the COBOL application when the MAP, OFFSET, or LIST compiler options are used.

Applications running in all environments, except MVS CICS and VSE CICS, use COBOL external data structures to share information between applications in the same run unit. Table 10 on page 16 shows the storage estimates for external data structures.

Table 10. COBOL External Storage Utilization in Non-CICS Environments

Function	Storage Required
VisualAge Generator Server for MVS, VSE, and VM control block	1KB
Environment is IMS/VS or IMS BMP	32KB
IMS conversational processing	SPA size plus 18 bytes
File type SEQ, VSAM, GSAM, SMSGQ, MMSGQ or EZEPRINT SEQ, GSAM	96 bytes/file

VisualAge Generator Server for MVS, VSE, and VM Dynamic Storage

When applications are running, VisualAge Generator Server for MVS, VSE, and VM allocates storage as shown in Table 11. The initial application of the run unit determines where the shared storage between VisualAge Generator Server for MVS, VSE, and VM and the generated COBOL application is allocated. If the initial application is generated with the /DATA=24 generation option or is link-edited with AMODE(24), this storage is allocated below the 16MB line. Otherwise, the storage is allocated with 31-bit addresses as shown in the following table:

Table 11. VisualAge Generator Server for MVS, VSE, and VM Dynamic Storage Utilization

Function	Storage Required	24- or 31-bit Addressing mode
Persistent dynamic storage pool. The pool is extended as needed in 32KB increments. Most transactions or jobs require only the initial allocation. Segmented transactions in MVS CICS or VSE CICS, or using a DB2 work database in IMS might require an extension.	32KB increment	31
MVS CICS, VSE CICS - service program dynamic storage stack	48KB	31
MVS CICS with IMS DM - DL/I buffers	64KB	31
VSE CICS with DL/I DOS/VS ¹ (maximum)	68KB	24
IMS, IMS BMP, MVS/TSO, MVS batch, VSE batch, VM CMS, VM batch - service program dynamic storage stack	48KB	24

Table 11. VisualAge Generator Server for MVS, VSE, and VM Dynamic Storage Utilization (continued)

Function	Storage Required	24- or 31-bit Addressing mode
IMS, MVS/TSO, MVS batch, VSE batch, VM CMS, VM batch - DL/I buffers for path calls, DL/I work database	64KB	24
IMS, MVS/TSO, MVS batch, VM CMS, VM batch - DL/I buffers for path calls, DL/I work database	64KB	31
IMS BMP - DL/I buffers for path calls, checkpoint input	96KB	24
IMS BMP - DL/I buffers for path calls, checkpoint input	96KB	31

Note:

1. For VSE environments, the IMSESA parameter in the installation options module is ignored because DL/I is restricted to run below the 16MB line.

Storage Requirements for MVS CICS and VSE CICS

Generated COBOL applications use the following MVS CICS and VSE CICS storage facilities:

Table 12. VisualAge Generator Server for MVS, VSE, and VM Use of MVS CICS and VSE CICS Storage Areas

Type of Storage	Function	Size
Transaction Work Area (TWA)	VisualAge Generator Server for MVS, VSE, and VM Control Block. Offset in TWA is specified in /TWAOFF generation option.	1KB
Shared Storage	Shared table contents Shared table control block	For each table, length of table contents plus 16 bytes for message table or 8 bytes for nonmessage table One 50-byte record per shared table
Temporary Storage Queue (Main or Auxiliary)	Save information during CONVERSE or XFER with map	6KB, plus length of all records and maps
COMMAREA	Segmented application	10 bytes

Table 12. VisualAge Generator Server for MVS, VSE, and VM Use of MVS CICS and VSE CICS Storage Areas (continued)

Type of Storage	Function	Size
COMMAREA	DXFR passing records	Length of record passed
COMMAREA	Calls with COMMPTR	4 times the number of parameters
COMMAREA	Calls with COMMDATA	Total length of all parameters
COMMAREA	XFER passing records	Length of record passed, plus 10
COMMAREA	Remote calls	Total length of all parameters, plus 12

Disk Storage Requirements for VisualAge Generator Server for MVS, VSE, and VM

The auxiliary disk storage space required to install the files for VisualAge Generator Server for MVS, VSE, and VM is approximately 2MB. Additional disk space for user applications can vary.

Work Database Space For Segmented Applications

The space required for saving application status across a terminal I/O operation in IMS, MVS CICS, or VSE CICS is the sum of all segmented applications' data areas (maps and records) plus 4KB per application for IMS or 6KB per application for MVS CICS or VSE CICS. In VSE CICS and MVS CICS, disk space is used only if auxiliary temporary storage is specified as the work database during application generation.

For example, suppose on IMS that application A has the following:

- Two 4KB records
- Two 512-byte maps
- 1KB of working storage
- 100 terminals running application A in segmented mode

The approximate required disk space for MVS CICS, VSE CICS, or for IMS with a DB2 work database is as follows:

$$(2 \times 4 \ 096 + 2 \times 512 + 1 \ 024 + 4 \ 096) \times 100 = 1 \ 433 \ 600$$

If you are using a DL/I work database with IMS, the storage required per terminal is inserted in 56KB increments to localize access for all segments accessed on a single-path call. An additional 56KB increment is required when

help maps or extended error screens are used. A good estimate for work database size is 112KB per active terminal.

Chapter 5. Installation Considerations for MVS, VSE, and VM Systems

The following sections describe installation considerations for VisualAge Generator Server for MVS, VSE, and VM.

Installation and Language-Dependent Options for MVS, VSE, and VM

The following are the language-dependent options required for MVS, VSE, and VM.

Table 13. Installation and Language-dependent Options for MVS, VSE, and VM

Question	Default	Your Selection
VisualAge Generator Server for MVS, VSE, and VM Runtime Default Options		
Default language code	ENU	_____
Bypass date edit on EOF	NO	_____
IMS/ESA ¹ installed	NO	_____
VisualAge Generator Server for MVS, VSE, and VM trace buffer size	64	_____
CICS ¹ temporary storage control interval size	16	_____
VisualAge Generator Server for MVS, VSE, and VM National Language Dependent Options (One code is needed for each national language you install. The default values vary for each language.)		
National language code (US English)		
Long Gregorian date format	MM/DD/YYYY	_____
Short Gregorian date format	MM/DD/YY	_____
Long Julian date format	YYYY-DDD	_____
Short Julian date format	YY-DDD	_____
Conversion table name	ELACNENU	_____
Positive response character string	YES	_____
Negative response character string	NO	_____
Decimal point character	.	_____
Numeric separator character	,	_____
Currency symbol	\$	_____
SQL host variable indicator	:	_____
SQL host column indicator	!	_____
National language code (Simplified Chinese)		
Long Gregorian date format	YYYY-MM-DD	_____
Short Gregorian date format	YY-MM-DD	_____

Table 13. Installation and Language-dependent Options for MVS, VSE, and VM (continued)

Question	Default	Your Selection
Long Julian date format	YYYY-DDD	_____
Short Julian date format	YY-DDD	_____
Conversion table name	ELACNCHS	_____
Positive response character string	YES	_____
Negative response character string	NO	_____
Decimal point character	.	_____
Numeric separator character	,	_____
Currency symbol	\$	_____
SQL host variable indicator	:	_____
SQL host column indicator	!	_____
National language code (Swiss German)	DES	_____
Long Gregorian date format	DD.MM.YYYY	_____
Short Gregorian date format	DD.MM.YY	_____
Long Julian date format	YYYY.DDD	_____
Short Julian date format	YY.DDD	_____
Conversion table name	ELACNDES	_____
Positive response character string	YES	_____
Negative response character string	NO	_____
Decimal point character	,	_____
Numeric separator character	.	_____
Currency symbol	\$	_____
SQL host variable indicator	:	_____
SQL host column indicator	!	_____
National language code (German)	DEU	_____
Long Gregorian date format	DD.MM.YYYY	_____
Short Gregorian date format	DD.MM.YY	_____
Long Julian date format	DDD/YYYY	_____
Short Julian date format	DDD/YY	_____
Conversion table name	ELACNDEU	_____
Positive response character string	YES	_____
Negative response character string	NO	_____
Decimal point character	,	_____
Numeric separator character	.	_____
Currency symbol	\$	_____
SQL host variable indicator	:	_____
SQL host column indicator	!	_____
National language code (Upper Case English)	ENP	_____
Long Gregorian date format	MM/DD/YYYY	_____
Short Gregorian date format	MM/DD/YY	_____
Long Julian date format	YYYY-DDD	_____
Short Julian date format	YY-DDD	_____
Conversion table name	ELACNENP	_____

Table 13. Installation and Language-dependent Options for MVS, VSE, and VM (continued)

Question	Default	Your Selection
Positive response character string	YES	_____
Negative response character string	NO	_____
Decimal point character	.	_____
Numeric separator character	,	_____
Currency symbol	\$	_____
SQL host variable indicator	:	_____
SQL host column indicator	!	_____
National language code (Spanish)	ESP	_____
Long Gregorian date format	DD/MM/YYYY	_____
Short Gregorian date format	DD/MM/YY	_____
Long Julian date format	DDD/YYY	_____
Short Julian date format	DDD/YY	_____
Conversion table name	ELACNESP	_____
Positive response character string	SI	_____
Negative response character string	NO	_____
Decimal point character	,	_____
Numeric separator character	.	_____
Currency symbol	\$	_____
SQL host variable indicator	:	_____
SQL host column indicator	!	_____
National language code (Japanese)	JPN	_____
Long Gregorian date format	YYY-MM-DD	_____
Short Gregorian date format	YY-MM-DD	_____
Long Julian date format	YYYY-DDD	_____
Short Julian date format	YY-DDD	_____
Conversion table name	ELACNJPN	_____
Positive response character string	YES	_____
Negative response character string	NO	_____
Decimal point character	.	_____
Numeric separator character	,	_____
Currency symbol	\$	_____
SQL host variable indicator	:	_____
SQL host column indicator	!	_____
National language code (Korean)	KOR	_____
Long Gregorian date format	MM/DD/YYYY	_____
Short Gregorian date format	MM/DD/YY	_____
Long Julian date format	DDD/YYYY	_____
Short Julian date format	DDD/YY	_____
Conversion table name	ELACNKOR	_____
Positive response character string	YES	_____
Negative response character string	NO	_____
Decimal point character	.	_____
Numeric separator character	,	_____

Table 13. Installation and Language-dependent Options for MVS, VSE, and VM (continued)

Question	Default	Your Selection
Currency symbol	\$	_____
SQL host variable indicator	:	_____
SQL host column indicator	!	_____
National language code (Brazilian Portuguese)	PTB	_____
Long Gregorian date format	DD/MM/YYYY	_____
Short Gregorian date format	DD/MM/YY	_____
Long Julian date format	DDD/YYYY	_____
Short Julian date format	DDD/YY	_____
Conversion table name	ELACNPTB	_____
Positive response character string	SIM	_____
Negative response character string	NAO	_____
Decimal point character	,	_____
Numeric separator character	.	_____
Currency symbol	\$	_____
SQL host variable indicator	:	_____
SQL host column indicator	!	_____

Note:

1. This environment is not supported in VM.

MVS Batch and MVS/TSO Installation Considerations

This section discusses some considerations for running VisualAge Generator applications in the MVS batch and MVS/TSO environments.

DL/I Considerations

If the installation has applications that gain access to DL/I databases, do the following:

1. Install the correct version of IMS. For more information on the correct version of IMS, see “Chapter 3. Software Requirements for VisualAge Generator Server for MVS, VSE, and VM” on page 7.
2. Define databases and PSBs to IMS as described in the IMS utilities reference document.
3. Follow the optional DL/I-related steps for VisualAge Generator Server installation as described in the *Program Directory for VisualAge Generator Server for MVS*.
4. The /DATA generation option defaults to /DATA=24 for non-CICS environments.

DB2 Considerations

If the installation has applications that gain access to relational databases, do the following:

1. Install the correct version of DB2. For more information on the correct version of DB2, see “Chapter 3. Software Requirements for VisualAge Generator Server for MVS, VSE, and VM” on page 7.

If the ANSI Standard SQL is being used, install a database system meeting the X3.135-1986 standard.

Refer to the /ANSISQL generation option in the *VisualAge Generator Generation Guide* for additional information about using an ANSI SQL database manager.

2. Create the tables in the relational database that the applications will access.
3. Follow the optional DB2-related steps for VisualAge Generator Server installation as described in the *Program Directory for VisualAge Generator Server for MVS*.
4. Define application plans as described in the resource definition and installation and operation guides.

Security Considerations

Security products such as Resource Access Control Facility (RACF) provide control of resources when running applications in MVS/TSO or MVS batch.

Making VisualAge Generator Server for MVS, VSE, and VM, COBOL, and the Generated Modules Available

If the generated application contains DL/I or SQL processes, the load library where the generated programs reside needs to have its contents accessible by one of the following methods:

- Adding the library to the STEPLIB concatenation list for the MVS/TSO logon procedure
- Using the ISPF LIBDEF ISPLLIB command. This is only required if you want to run the applications from within ISPF.
- Using the TSOLIB command provided by TSO

If the generated application does not contain DL/I or SQL processes, the MVS/TSO CALL command that is used to start the application establishes a temporary TASKLIB search chain. This TASKLIB search chain is searched when locating or loading the load module. The TASKLIB is only available during the CALL. Therefore, if neither DL/I nor DB2 is used and all the programs (applications, print services programs, map group format modules, table programs, and non-VisualAge Generator programs) for the run unit are in the same load library, the above techniques are not required.

For all types of applications, the VisualAge Generator Server runtime libraries and the LE runtime libraries must be made available using one of the above techniques. The system administrator might have included these libraries in the STEPLIB concatenation list when VisualAge Generator Server was installed and customized.

MVS CICS Installation Considerations

This section discusses some general considerations for running VisualAge Generator applications in the MVS CICS environment.

CICS/ESA Considerations

CICS/ESA supports DBCS names in COBOL applications. By default, the VisualAge Generator Server Application Generator for MVS assigns alias names for DBCS names in applications generated for CICS. The /CICSDBCS generation option can be specified to prevent the generator from creating aliases for DBCS names.

DL/I Considerations

If the installation has applications that gain access to DL/I databases, you must do the following:

1. Install the correct version of IMS. For more information on the correct version of IMS, see “Chapter 3. Software Requirements for VisualAge Generator Server for MVS, VSE, and VM” on page 7.
2. Define databases and PSBs to IMS as described in the IMS utilities reference document.
3. Follow the optional DL/I-related steps for VisualAge Generator Server installation as described in the *Program Directory for VisualAge Generator Server for MVS*.
4. Add DL/I support to MVS CICS and define databases and PSBs to MVS CICS as described in the resource definition and installation and operation guides or in the IMS database control guide.

DB2 Considerations

If the installation has applications that gain access to relational databases, do the following:

1. Install the correct version of DB2. For more information on the correct version of DB2, see “Chapter 3. Software Requirements for VisualAge Generator Server for MVS, VSE, and VM” on page 7.

If the ANSI Standard SQL is being used, install a database system meeting the X3.135-1986 standard.

Refer to the /ANSISQL generation option in the *VisualAge Generator Generation Guide* for additional information about using an ANSI SQL database manager.

2. Create the tables in the relational database that the applications will gain access to.
3. Follow the optional DB2-related steps for VisualAge Generator Server installation as described in the *Program Directory for VisualAge Generator Server for MVS*.
4. Add DB2 support to MVS CICS and define application plans to MVS CICS as described in the DB2 system administration guides.

Security Considerations

MVS CICS provides access control to resources (such as data files and programs) and transactions. This access can be controlled by the user or by the terminal.

MVS CICS resources (such as data files, programs, destinations, journals, and temporary storage) can be assigned a security lock value. MVS CICS users are assigned one or more key values. If a user is running an MVS CICS transaction that is defined for resource security checking, the user's keys are checked every time a resource is requested. If the user does not have a key that matches the lock, access is denied by ending the transaction with an AEY7 ABEND code.

Monitoring and Tuning

Use the CICS/ESA monitoring facility to get information about CICS/ESA tasks.

Refer to the performance guide for your release of CICS for more information.

MVS CICS Utilities

In the MVS CICS environment, VisualAge Generator Server for MVS, VSE, and VM includes a set of utilities to assist in managing the error diagnosis and control facilities of the VisualAge Generator Server for MVS, VSE, and VM runtime environment. These utilities are VisualAge Generator applications. See "Using the CICS Utilities Menu" on page 181 for more information about these utilities.

Client/Server Processing Considerations

VisualAge Generator applications can use the benefits of client/server processing in the CICS environments of MVS, VSE, OS/2, and AIX. Client/server applications are developed like any other VisualAge Generator

application. Client/server processing is built on the CALL, CREATX, and file I/O processing options. Using VisualAge Generator definition facilities, an application can be defined so that it calls an application on a remote system, starts an asynchronous application on a remote system, or gains access to a file on a remote system. VisualAge Generator GUI applications can start generated COBOL server applications using the CALL statement. Refer to the *VisualAge Generator Client/Server Communications Guide* for additional information about client/server applications.

Using the /DATA Generation Option

Specify the /DATA=24 generation option on generated COBOL applications to enable calls from the generated application to applications using 24-bit addresses, as long as the length of the COBOL dynamic storage (as defined in the COBOL working-storage section) required for the application is less than 64KB. Programs whose dynamic storage requirements are greater than 64KB must be compiled with the /DATA=31 generation option. Otherwise, COBOL ends the program with a 1009 ABEND code.

Note: The procedures shipped with VisualAge Generator Server for MVS, VSE, and VM enable the /DATA generation option to control the value for the COBOL DATA compiler option. /DATA=31 is the default for the MVS CICS environment.

Modifying MVS CICS Resource Tables

MVS CICS uses tables to identify startup parameters, transactions, programs, files, databases, transient data destinations, and system locations for proper operation. The application developer must add or modify these tables to correctly identify all objects to be used in the new or changed application. The MVS CICS tables are compiled as assembler programs and stored in a runtime library. Some tables can also be maintained through an online facility as described in the MVS CICS resource definition online document. CICS/ESA requires that the online facility be used instead of PPT and PCT entries.

To generate model resource definition online (RD) program and transaction definitions, specify /CICSENTRIES=RDO.

The MVS CICS system initialization table needs to include EXEC=YES.

Make the INQUIRESET programs available to all VisualAge Generator applications.

Add any transaction that invokes an application that uses DB2 to the resource control table (RCT) with the appropriate plan name.

Using Spool Files

To use the spool files, include the SPOOL=YES parameter in the System Initialization Table (SIT).

Terminal Considerations

Terminals used with VisualAge Generator must have their alternate screen size either specified correctly in the alternate screen parameter of the DFHTCT macro, or omitted so the default of the primary screen size is used. An alternate screen size specification of (0,0) is not valid.

Any terminal defined as UCTRAN and used for running pseudoconversational transactions might give different results than a terminal that is not defined as UCTRAN.

Any terminal used in an application that is the target of an XFER must have TRMSTAT=TRANSCEIVE specified in its DFHTCT TYPE=TERMINAL entry.

Temporary Storage

Temporary storage queues used by VisualAge Generator Server for MVS, VSE, and VM must be defined as nonrecoverable. These queues start with X'EE'.

IMS Installation Considerations

This section discusses some general considerations for running VisualAge Generator applications in the IMS environment.

IMS/ESA Exploitation

The procedures shipped with VisualAge Generator Server for MVS, VSE, and VM cause the generated COBOL applications to be compiled with the /DATA=24 generation option and linked in AMODE(31) and RMODE(ANY). If the application calls another application that is linked with AMODE(24), then the /DATA=24 generation option is required.

You can link the generated COBOL application to run below the 24-bit line. However, if AMODE(24) is used to link the application, you must use /DATA=24 generation option for the following situations:

- For an application that calls an application or program that is linked as AMODE(24)
- For the first application in the run unit, if any generated application in the run unit is linked as AMODE(24) or if a non-VisualAge Generator application that uses DL/I is linked as AMODE(24)

- For a table or mapping services program, if any application being used is linked as AMODE(24)

DB2 Considerations

If the installation has applications that gain access to relational databases, do the following:

1. Install the correct version of DB2. For more information on the correct version of DB2, see “Chapter 3. Software Requirements for VisualAge Generator Server for MVS, VSE, and VM” on page 7.

If the ANSI Standard SQL is being used, install a database system meeting the X3.135-1986 standard.

Refer to the /ANSISQL generation option in the *VisualAge Generator Generation Guide* for additional information about using an ANSI SQL database manager.

2. Create the tables in the relational database that the applications will gain access to.
3. Follow the optional DB2-related steps for VisualAge Generator Server installation as described in the *Program Directory for VisualAge Generator Server for MVS*.
4. Add DB2 support to IMS and define application plans to IMS as described in the DB2 system administration guide.

Security Considerations

Resource Access Control Facility (RACF) can be used to control users authority to each transaction.

Monitoring and Tuning

Potential performance problems can be tracked before they occur by checking processing statistics on a regular basis. The following are some of the statistics to monitor:

- Use the IMS DC monitor facilities to check transaction utilization. Consider preloading applications or groups of applications that are frequently used.
- Use the IMS database monitor facilities to check how effectively the databases are performing and using space.

Use the following tools to monitor IMS performance:

- The IMS Performance Analysis and Reporting System (IMSPARS, program number 5798-CQP).

This tool presents information on transaction transit times, IMS resource usage, and IMS resource availability, and detailed reports tracing individual transaction and database change activity. These reports are based on the contents of the IMS log data set.

- The Resource Measurement Facility (RMF) II
This tool collects information about processor, channel, and I/O device utilization.
- The DB Tools product (program number 5668-856)
This tool provides information to help improve access efficiency and space utilization.

Refer to the IMS system administration document and the database administration guide for the release of IMS for additional information on monitoring the IMS online system and DL/I databases.

IMS System Definition

If you plan to use IMS, define all PSBs and transactions in the IMS system definition. In addition, define DL/I application databases.

IMS Control Region

You might need to review the values for the following:

- PSB work area pool (PSBW parameter)
- FORMAT pool (FBP parameter)
- MFS test area (MFS parameter)
- Communications input/output area (TPDP parameter)

In addition, if a DL/I work database is used, the work database would need to be added to either the control region JCL or to the dynamic allocation table.

Work Database

The work database is used to save the status of a VisualAge Generator application during a CONVERSE process option, and to pass information during certain types of program-to-program message switches. The work database can be either a DL/I database or a DB2 table. The application developer specifies the /WORKDB generation option when generating an application to determine which type of database is to be used. A DL/I or DB2 work database is used only for VisualAge Generator transaction applications that are generated for the IMS/VS target environment. In general, a DL/I work database performs better than a DB/2 work database.

Multiple DL/I or DB2 work databases can be installed. Use separate databases for each application system to improve performance or data availability.

DL/I Work Database Considerations

If you plan to use a DL/I implementation for the work database, you might need to tailor the database description (DBD) before running the job that creates and initializes the DL/I work database.

DB2 Work Database Considerations

If you plan to use a DB2 implementation for the work database, review the database definition before running the job that initializes the DB2 work database. A DB2 synonym needs to be created for each user and application gaining access to the DB2 work database.

The DB2 work database requires a 32KB page size. If a DB2 work database is used, you might need to increase the allocation of the 32KB buffers. To increase the allocation of buffers, modify and assemble the DB2 parameter module (default is DSNZPARM). Refer to the DB2 documents for the system for additional information.

If you select DB2, a DB2 application plan for each transaction is needed even if the VisualAge Generator application itself does not require DB2.

If you select DB2 and if VisualAge Generator Server for MVS, VSE, and VM needs maintenance applied to the module that handles the DB2 work database access, bind the application plans again for all transactions that use this database.

There are also considerations with the DB2 authorization used by the IMS program that is gaining access to the DB2 work database. For example, authorization needs to be granted to LTERM and a synonym needs to be created.

VSE Batch Installation Considerations

This section discusses some general considerations for running VisualAge Generator applications in the VSE batch environment.

DL/I Considerations

If the installation has applications that gain access to DL/I databases, do the following:

1. Install the required DL/I licensed program. For more information on the correct version of DL/I, see “Chapter 3. Software Requirements for VisualAge Generator Server for MVS, VSE, and VM” on page 7.

2. Define databases and PSBs to DL/I DOS/VS as described in the DL/I DOS/VS resource definition and utilities document.
3. Follow the optional DL/I-related steps for VisualAge Generator Server installation as described in the *Program Directory for VisualAge Generator Server for VSE*.
4. Specify the /DATA=24 generation option for any application that uses DL/I.

DB2/VSE Considerations

If the installation has applications that gain access to relational databases, do the following:

1. Install the correct version of DB2/VSE. For more information on the correct version of DB2/VSE, see “Chapter 3. Software Requirements for VisualAge Generator Server for MVS, VSE, and VM” on page 7.
If ANSI Standard SQL is used, install a database system meeting the X3.135-1986 standard.
Refer to the /ANSISQL generation option in the *VisualAge Generator Generation Guide* for additional information about using an ANSI SQL database manager.
2. Create the tables in the relational database that the applications will gain access to.
3. Follow the optional DB2/VSE-related steps for VisualAge Generator Server installation as described in the *Program Directory for VisualAge Generator Server for VSE*.

Making VisualAge Generator Server for MVS, VSE, and VM, COBOL, and the Generated Modules Available

If the generated application contains DL/I or SQL processes, the sublibrary where the generated programs reside needs to have its contents accessible by one of the following methods:

- Adding the sublibrary to the phase LIBDEF concatenation list for the VSE startup job
- Ensuring the sublibrary is in the phase LIBDEF of the VSE batch job

VSE CICS Installation Considerations

This section discusses some general considerations for running VisualAge Generator-generated applications in the VSE CICS environment.

DL/I Considerations

If the installation has applications that gain access to DL/I databases, do the following:

1. Install the required DL/I licensed program. For more information on the correct version of DL/I, see “Chapter 3. Software Requirements for VisualAge Generator Server for MVS, VSE, and VM” on page 7.
2. Define databases and PSBs to DL/I DOS/VS as described in the DL/I DOS/VS resource definition and utilities document.
3. Follow the optional DL/I-related steps for VisualAge Generator Server installation as described in the *Program Directory for VisualAge Generator Server for VSE*.
4. Add DL/I support to VSE CICS and define databases and PSBs to VSE CICS as described in the resource definition and installation and operation guides.

DB2/VSE Considerations

If the installation has applications that gain access to relational databases, do the following:

1. Install the correct version of DB2/VSE. For more information on the correct version of DB2/VSE, see “Chapter 3. Software Requirements for VisualAge Generator Server for MVS, VSE, and VM” on page 7.
If ANSI Standard SQL is used, install a database system meeting the X3.135-1986 standard.
Refer to the /ANSISQL generation option in the *VisualAge Generator Generation Guide* for additional information about using an ANSI SQL database manager.
2. Create the tables in the relational database that the applications will gain access to.
3. Follow the optional DB2/VSE-related steps for VisualAge Generator Server installation as described in the *Program Directory for VisualAge Generator Server for VSE*.
4. Add DB2/VSE support to VSE CICS as described in the DB2/VSE system administration guides.

Security Considerations

VSE CICS provides access control to resources (such as data files and programs) and transactions. This access can be controlled by user or by terminal.

VSE CICS resources (such as data files, programs, destinations, journals, and temporary storage) can be assigned a security lock value. VSE CICS users are

assigned one or more key values. If a user is running a VSE CICS transaction that is defined for resource security checking, the user's keys are checked every time a resource is requested. If the user does not have a key that matches the lock, access is denied by ending the transaction with an AEY7 ABEND code.

Monitoring and Tuning

For CICS/VSE, the CICS/VSE Performance Analysis Reporting System (CICSPARS/VSE) can be used to monitor performance. This tool provides information about current tasks, snapshots of the current system status, and information about the CICS/VSE dynamic storage area, and the ability to display CICS/VSE storage.

Refer to the performance guide for the release of VSE CICS for more information.

VSE CICS Utilities

In the VSE CICS environment, VisualAge Generator Server for MVS, VSE, and VM includes a set of utilities to assist in managing the error diagnosis and control facilities of the VisualAge Generator Server for MVS, VSE, and VM runtime environment. These utilities are VisualAge Generator applications. See "Using the CICS Utilities Menu" on page 375 for more information about these utilities.

Client/Server Processing Considerations

VisualAge Generator applications can use the benefits of client/server processing in the CICS environments of VSE, MVS, OS/2, and AIX. Client/server applications are developed like any other VisualAge Generator application. Client/server processing is built on the CALL, CREATX, and file I/O processing options. Using VisualAge Generator definition facilities, an application can be defined so that it calls an application on a remote system, starts an asynchronous application on a remote system, or gains access to a file on a remote system. VisualAge Generator GUI applications can start generated COBOL server applications using the CALL statement. Refer to the *VisualAge Generator Client/Server Communications Guide* for additional information about client/server applications.

Using the /DATA=24 Generation Option

The /DATA=24 generation option can be specified on generated COBOL applications to enable calls from the generated application to applications using 24-bit addresses, as long as the length of the COBOL dynamic storage (as defined in the COBOL working-storage section) required for the application is less than 64KB. Programs whose dynamic storage requirements

are greater than 64KB must be generated with the /DATA=31 option. Otherwise, COBOL ends the program with a 1009 ABEND code.

Modifying VSE CICS Resource Tables

VSE CICS uses tables to identify startup parameters, transactions, programs, files, databases, transient data destinations, and system locations for proper operation. The application developer must add or modify these tables to correctly identify all objects to be used in the new or changed application. The VSE CICS tables are compiled as assembler programs and stored in a runtime library. Some tables can also be maintained through an online facility as described in the VSE CICS resource definition online document.

To generate model resource definition online (RD) programs and transaction definitions, specify /CICSENTRIES=RDO. To generate model PPT and PCT table entries, specify /CICSENTRIES=MACRO.

The VSE CICS system initialization table needs to include EXEC=YES.

Make the INQUIRESET programs available to all VisualAge Generator applications.

Using Spool Files

To use the spool files, include the SPOOL=YES parameter in the System Initialization Table (SIT).

Terminal Considerations

Terminals used with VisualAge Generator must have their alternate screen size either specified correctly in the alternate screen parameter of the DFHTCT macro, or omitted so the default of the primary screen size is used. An alternate screen size specification of (0,0) is not valid.

Any terminal defined as UCTRAN and used for running pseudoconversational transactions might give different results than a terminal that is not defined as UCTRAN.

Any terminal used in an application that is the target of an XFER must have TRMSTAT=TRANSCIVE specified in its DFHTCT TYPE=TERMINAL entry.

Temporary Storage

Temporary storage queues used by VisualAge Generator Server for MVS, VSE, and VM must be defined as nonrecoverable. These queues start with X'EE'.

VM Batch and VM CMS Installation Considerations

This section discusses some considerations for running VisualAge Generator applications in the VM batch and VM CMS environments.

SQL/DS Considerations

If the installation has applications that gain access to relational databases, do the following:

1. Install the correct version of SQL/DS. For more information on the correct version of SQL/DS, see “Chapter 3. Software Requirements for VisualAge Generator Server for MVS, VSE, and VM” on page 7.

If ANSI Standard SQL is used, install a database system meeting the X3.135-1986 standard.

Refer to the /ANSISQL generation option in the *VisualAge Generator Generation Guide* for additional information about using an ANSI SQL database manager.

2. Create the tables in the relational database that the applications will gain access to.
3. Follow the optional SQL/DS-related steps for VisualAge Generator Server installation as described in the *Program Directory for VisualAge Generator Server for VM*.

Security Considerations

Security products such as Resource Access Control Facility (RACF) provide control of resources when running applications in VM CMS or VM batch.

Chapter 6. Customizing VisualAge Generator Server for MVS, VSE, and VM

Customization of VisualAge Generator Server for MVS, VSE, and VM consists of tailoring the product for use.

Before starting the customization process, determine the following:

1. The target environments that application developers specify during generation
2. Whether the applications use hierarchical or relational application databases, or both
3. For IMS/VS target environment, determine the following:
 - a. Whether to use a hierarchical or relational database implementation for the VisualAge Generator Server for MVS, VSE, and VM work database. See “Work Database” on page 31 for additional information on choosing between a hierarchical or relational database implementation for the work database.
 - b. The types of terminals that are used with the generated applications. See “Considerations for Message Format Services in IMS” on page 105 for the information needed to define terminals using the /MFSDEV, /MFSEATTR, and /MFSIGNORE generation options. You need information from the IMS system definition and samples of MFS from existing applications to help define this information.
4. For the MVS/TSO target environment, which logon procedures are used by the end users of VisualAge Generator applications.
5. The national language support requirements

General Customization Considerations for MVS, VSE, and VM Systems

The following sections discuss some general considerations for running VisualAge Generator-generated applications in the supported MVS, VSE, and VM environments.

Security Considerations

VisualAge Generator Server for MVS, VSE, and VM does not provide security services. Standard system or database manager security functions can be used with generated COBOL applications in the same way that they are used with customer-developed COBOL applications.

For example, if the applications use DB2, define DB2 application plans and give run authority to those users that are authorized to use the applications associated with the plan. The Resource Access Control Facility (RACF*) can also be used to grant users authority to read or update files.

Performance Considerations

Other chapters in this book provide detailed information on considerations that affect performance. See the following chapters for information on these performance-related topics and others:

Performance Topic	Where to Find Info
Generation and compile options	<ul style="list-style-type: none"> • “Chapter 7. General System Considerations for MVS Systems” on page 45 • “Chapter 30. General System Considerations for VSE Systems” on page 291 • “Chapter 47. General System Considerations for VM Systems” on page 467
Placing VisualAge Generator Server product and generated application modules or phases in memory	<ul style="list-style-type: none"> • “Chapter 7. General System Considerations for MVS Systems” on page 45 • “Chapter 30. General System Considerations for VSE Systems” on page 291
Residency considerations in MVS CICS and VSE CICS	<ul style="list-style-type: none"> • “Chapter 8. System Considerations for MVS CICS” on page 55 • “Chapter 11. System Considerations for IMS” on page 87 • “Chapter 31. System Considerations for VSE CICS” on page 303
Preload considerations in IMS	<p>“Chapter 18. Preparing and Running Generated Programs in IMS/VS and IMS BMP” on page 165</p>

Performance Topic	Where to Find Info
Work database considerations in MVS CICS, VSE CICS, and IMS	<ul style="list-style-type: none"> • “Chapter 8. System Considerations for MVS CICS” on page 55 • “Chapter 11. System Considerations for IMS” on page 87 • “Chapter 31. System Considerations for VSE CICS” on page 303
Monitoring and tuning tools	<ul style="list-style-type: none"> • “Chapter 8. System Considerations for MVS CICS” on page 55 • “Chapter 11. System Considerations for IMS” on page 87

Customizing JCL Procedures

VisualAge Generator Server for MVS, VSE, and VM includes procedures used for preparing generated applications for running. These procedures can be customized to meet the data set naming conventions. Refer to the *VisualAge Generator Generation Guide* for additional information on customizing JCL procedures.

Modifying the COBOL Working Storage Initialization Option

In the non-CICS environments, generated COBOL applications rely on COBOL working storage being initialized to binary zeros to determine whether VisualAge Generator Server for MVS, VSE, and VM is initialized. For LE, this is done by specifying `STORAGE=((00))` in the `CEEDOPT` CSECT. The modified runtime options module (or modules) must be in a library allocated to the `STEPLIB`, placed in the link pack area or in a library managed by the Virtual Lookaside Facility and Library Lookaside features of MVS/ESA for each non-CICS MVS environment, or must be in the shared virtual area list for VSE batch. For VM, the modules are compiled and linked into the VisualAge Generator runtime services `LOADLIB` as part of installation tailoring. If the module(s) are put in a separate library, the library must precede the library containing the unmodified modules.

Using Generated Programs with PL/I Programs

If PL/I programs are used with generated COBOL applications in a non-CICS environment, you must generate the COBOL application to invoke the PL/I program using a static COBOL call. This requires the PL/I programs to be linked with the COBOL application in the same load module. Before attempting to develop the COBOL application, refer to the *VisualAge Generator Client/Server Communications Guide* for additional information on defining linkage tables and specifying linkage editor control statements.

If PL/I programs are used with generated COBOL applications in the MVS CICS and VSE CICS environments, you must generate the COBOL application to call the PL/I program using the CICS LINK command. This is the default linkage for the MVS CICS and VSE CICS environments. The calling and called programs must not be linked together for the MVS CICS and VSE CICS environments.

Part 2. Administering on MVS Systems

Chapter 7. General System	
Considerations for MVS Systems	45
Considerations that Affect Performance	45
Generation and Compiler Options	45
Modules in Memory	46
Files and Databases	47
Defining and Loading VSAM Program Data Files	47
Defining VSAM Data Sets	47
Defining an Alternate Index	49
Loading Data in the Files	50
Support for DBCS Terminals	51
Extended Addressing Considerations for VisualAge Generator Server	51
Database Considerations	52
Preparing Programs	52
Checking Access Authorization	52
Fold Tables.	53
Backing Up Data.	53
Customizing VisualAge Generator Server	53
Chapter 8. System Considerations for MVS CICS	55
Required File Descriptions	55
Segmented and Nonsegmented Processing	56
Using Transient Data Queues for Printing in MVS CICS	57
Processing a DISPLAY Process Option for a Printer Map	57
Specifying Print Destination	58
MVS CICS Terminal Printing.	58
Special Parameter Group for the FZETPRT Program	59
PRTBUF Parameter	60
PRTMPP Parameter	61
PRTTYP Parameter	61
FORMFD Parameter.	62
CICS Entries for FZETPRT (DBCS only)	62
Using the New Copy Function	63
Recovery and Database Integrity Considerations	63
Specifying Recovery Options in the CICS Tables	64
Determining When Database Changes Are Committed	64
Considerations that Affect Performance	65
Residency (Modules in Memory) Considerations	65
Virtual Storage Considerations and Residency	66
Pseudoconversational Programs and Residency	67
Work Database Temporary Storage Queue Considerations	67
Database Temporary Storage Size	68
Conversational and Pseudoconversational Processing	68
Terminal Printing	69
Monitoring and Tuning the System.	69
Using and Allocating Data Files in MVS CICS and VSE CICS.	69
Defining and Loading VSAM Data Files	69
Adding the Job Control Statements	69
Adding the File Name to the CICS File Control Table	70
Using Remote Files	71
Defining Transient Data Queues.	72
Defining Intrapartition Transient Data	72
Defining Extrapartition Transient Data	73
Considerations for Using DB2 in MVS CICS	73
Associating DB2 Databases with CICS Transactions	74
Recovery and Database Integrity Considerations	74
Considerations for Using DL/I in MVS CICS.	74
Recovery and Database Integrity Considerations	75
Chapter 9. System Considerations for MVS/TSO	77
Required File Descriptions	77
Printing Considerations for MVS/TSO	78
Making VisualAge Generator Server, COBOL, and the Generated Modules Available	78
Recovery Considerations	79
Using VSAM Program Data Files in MVS/TSO	80
Considerations for Using DB2 in MVS/TSO	80

Recovery and Database Integrity Considerations	81	Processing Modes	90
Considerations for Using DL/I in MVS/TSO	81	Printing Considerations for IMS.	90
DL/I Support in Generated Programs	81	Recovery and Database Integrity Considerations	90
Defining the Program Specification Block (PSB).	81	Considerations that Affect Performance	91
Recovery and Database Integrity Considerations	82	Residency Considerations and the IMS Preload Function.	92
Chapter 10. System Considerations for MVS Batch	83	Preloading VisualAge Generator Server Modules	93
Required File Descriptions	83	Loading VisualAge Generator Server Modules into the Link Pack Area	94
Recovery Considerations	83	Preloading Generated Programs.	94
Using VSAM Program Data Files in MVS Batch.	84	Database Performance	94
Considerations for Using DB2 in MVS Batch	84	Limiting MFS Control Blocks.	95
Recovery and Database Integrity Considerations	84	Monitoring and Tuning the IMS System	95
Considerations for Using DL/I in MVS Batch.	85	Considerations for Using DB2 in IMS	96
Defining the Program Specification Block (PSB).	85	Recovery and Database Integrity Considerations	96
Recovery and Database Integrity Considerations	85	Checking Authorization	96
Performance Considerations for MVS Batch.	86	Considerations for Using DL/I in IMS	97
Runtime JCL	86	Recovery and Database Integrity Considerations	97
Chapter 11. System Considerations for IMS	87	Maintaining the Work Database in IMS	98
Required File Descriptions	87	Deleting Old Records from the Work Database	98
Defining the Program Specification Block (PSB).	88	DL/I Work Database	98
Migrating from CSP/370RS V1R1	90	DB2 Work Database	99
		Expanding the Work Database	101
		DL/I Work Database	101
		DB2 Work Database	101
		Supporting Multiple Work Databases	103
		DL/I Work Databases	104
		DB2 Work Databases	104
		Considerations for Message Format Services in IMS	105

Chapter 7. General System Considerations for MVS Systems

This chapter describes the system requirements and considerations for administering VisualAge Generator Server in all of the supported MVS environments.

The following information is discussed:

- Considerations that affect performance
- Defining and loading VSAM program data files
- Support for DBCS terminals
- Extended addressing considerations for VisualAge Generator Server
- DB2* considerations
- Fold tables
- Backing up data
- Customizing VisualAge Generator Server

Considerations that Affect Performance

Specifying certain generation and compiler options and making reentrant programs resident in memory can affect the performance of VisualAge Generator programs.

Generation and Compiler Options

Specifying the following generation options might improve runtime performance:

- /INEDIT=INONLY
- /MATH=COBOL
- /NODEBUGTRACE
- /NOINITADDWS
- /NOINITRECD
- /NOLEFTJUST
- /NONULLFILL
- /NONUMOVFL
- /NOSETFULL
- /NOVALIDMIX

Specifying the following compiler options might improve runtime performance:

- NOFDUMP
- NOSSRANGE
- NOTEST
- OPT

Note: Refer to the *VisualAge Generator Generation Guide* for details on these options.

Modules in Memory

Placing load modules in memory can improve performance by reducing the number of I/O operations (EXCPs). Load modules can be placed in memory by using the features of MVS or the features of the environment in which you are running. Two general MVS methods for improving performance are described in the remainder of this section. Refer to the appropriate performance consideration sections for more detailed information about improving performance in a particular runtime environment.

General MVS/ESA* methods to place load modules in memory are listed below:

- Place modules in the link pack area (LPA). Some of the modules that are shipped with VisualAge Generator Server are reentrant and can be placed in the LPA. Refer to the *Program Directory for VisualAge Generator Server for MVS (GI10-0812)* for information about modules that are reentrant and LPA eligible.

Generated programs, online print service programs, map group format modules, and shared tables are also reentrant and can be included in the LPA.

- Manage the VisualAge Generator Server data sets and the data sets containing the generated programs, online print services programs, map group format modules, and shared tables using the Virtual Lookaside Facility (VLF) and the Library Lookaside (LLA) features of MVS/ESA. These features can place both the load modules and the partitioned data set (PDS) directories in memory. Refer to the *MVS/ESA Initialization and Tuning* manual for more information about using these features.

Note: The STEPLIB and ISPLLIB libraries are searched first. For the MVS/ESA methods, the load module (for LPA) or the data set (for VLF/LLA) cannot be contained in the STEPLIB or ISPLLIB concatenation list.

Files and Databases

Standard tuning techniques (such as buffering) can be used with files and databases used by generated COBOL programs.

Defining and Loading VSAM Program Data Files

This section describes how to define and load VSAM data sets for use as program data files in the MVS CICS, MVS/TSO, or MVS batch environments. The section contains the following information:

- Defining VSAM data sets
- Defining an alternate index
- Loading data into the files

Defining VSAM Data Sets

VSAM data files can be serial (ESDS), relative (RRDS), or indexed (KSDS) files. Use the IDCAMS program to define a user VSAM data file. Figure 1 on page 48 shows example JCL that can be used to define the VSAM files.

```

//DEFVSAM JOB ...
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *

/* THE FOLLOWING SAMPLE DEFINES A */
/* VSAM INDEXED FILE */

DEFINE CLUSTER (NAME(ELA1.USER.KSDS) -
              VOL(xxxxxx) -
              CYLINDERS(pp ss) -
              KEYS(1 d) -
              RECORDSIZE(aaa mmm) -
              INDEXED)

/* THE FOLLOWING SAMPLE DEFINES A VSAM */
/* NUMBERED RELATIVE RECORD FILE */

DEFINE CLUSTER (NAME(ELA1.USER.RRDS) -
              VOL(xxxxxx) -
              CYLINDERS(pp ss) -
              RECORDSIZE(aaa mmm) -
              NUMBERED)

/* THE FOLLOWING SAMPLE DEFINES A VSAM */
/* ESDS FILE */

DEFINE CLUSTER (NAME(ELA1.USER.ESDS) -
              VOL(xxxxxx) -
              CYLINDERS(pp ss) -
              RECORDSIZE(aaa mmm) -
              NONINDEXED)

```

where:

- xxxxxx** Specifies a valid volume serial number
- pp** Specifies the primary number of cylinders to be allocated
- ss** Specifies the secondary number of cylinders to be allocated
- l** Specifies the length of the key
- d** Specifies the offset of the key
- aaa** Specifies the desired average record length
- mmm** Specifies the maximum record length

Figure 1. Defining VSAM Data Files

Defining an Alternate Index

An alternate index provides you with another way of gaining access to the records in a given KSDS file. Using a secondary key eliminates the need for you to keep several copies of the same information organized in different ways for different programs.

To gain access from an alternate index to the file through its prime index (base cluster), you must define a path to it. The path sets up an association between the alternate index and the base cluster, allowing the records in the data set to be available to you in different sequences. The alternate index is built after the base cluster is defined.

Figure 2 shows example IDCAMS definition commands for the base cluster and the alternate index cluster for an indexed file.

```
DEFINE CLUSTER (NAME(VSAM.KSDS.BASE.FILE) -  
  VOLUMES(ELAVOL) -  
  CYLINDERS(pp ss) -  
  KEYS(1 d) -  
  RECORDSIZE(aaa mmm) -  
  INDEXED)  
DEFINE ALTERNATEINDEX (NAME(VSAM.KSDS.ALT.INDEX) -  
  KEYS(1 d) -  
  CYLINDERS(pp ss) -  
  RELATE(VSAM.KSDS.BASE.FILE) -  
  VOLUMES(ELAVOL))  
DEFINE PATH(NAME(VSAM.KSDS.ALT.INDEX.PATH) -  
  PATHENTRY(VSAM.KSDS.ALT.INDEX))  
BLDINDEX INDATASET(VSAM.KSDS.BASE.FILE) -  
  OUTDATASET(VSAM.KSDS.ALT.INDEX)
```

where:

- pp** Specifies the primary number of cylinders to be allocated
- ss** Specifies the secondary number of cylinders to be allocated
- l** Specifies the key length
- d** Specifies the key displacement
- aaa** Specifies the desired average record length
- mmm** Specifies the maximum record length

Figure 2. Defining the Base Cluster and the Alternate Index Cluster

Loading Data in the Files

If you are using a VSAM indexed file (KSDS) and you want to open it for input only, initialize the file with at least one record. The file must have at least one record because a VSAM restriction prevents a file from being opened for input if the file is empty. While an empty file might be opened for output or both input and output, it must contain data to be opened for input.

There are several ways that you can put data into a file. One way is to create a VisualAge Generator program that uses an ADD process option to add records to an empty serial file. Once the program ends, you can use the IDCAMS REPRO command to copy the serial file into an indexed file.

Another way is to write a program that uses an ADD process option to add records to an empty indexed file. You must close the file in order to make the new records accessible.

Another way to initialize a VSAM KSDS file is to use a utility program shipped with the VisualAge Generator Server product. This utility can be used to initialize the key of a VSAM KSDS file. Figure 3 shows how to initialize a VSAM KSDS file by setting the key to hexadecimal zeros.

```
//LOAD      JOB...
//JOBLIB    DD DSN=ELA.V1R2M0.SELALMD,DISP=SHR
//INITK     EXEC PGM=FZEZREBO,PARM='I,KSDS'
//SYSPRINT  DD SYSOUT=A,DCB=(LRECL=121,BLKSIZE=121,RECFM=FB)

//KSDS     DD DSN=USER.KSDS,DISP=SHR
//SYSIN     DD DUMMY
```

Figure 3. Initializing a VSAM KSDS File

You can also use the IDCAMS utility to load initial data into an indexed file. Figure 4 shows an example of loading data into a VSAM KSDS file. The data contained in the USER.KSDS.INPUT file is loaded into the USER.KSDS data set.

```
//JOB KSDSLOAD
//LOAD EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
      REPRO INDATASET('USER.KSDS.INPUT') OUTDATASET('USER.KSDS')
/*
/&
```

Figure 4. Loading a VSAM KSDS File

Support for DBCS Terminals

VisualAge Generator Server provides support for the IBM Personal System/55 and the IBM 5550 family of terminals (emulating an IBM 3270 device). In addition to the basic hardware, this support uses character set F8 and four hardware attributes for double-byte character set (DBCS). The extended attributes are shift-out (SO) and shift-in (SI) enable, field outlining, color, and extended highlighting.

For the MVS CICS, MVS/TSO, VSE, and VM CMS environments, VisualAge Generator Server sends hardware attributes to the terminal only if the terminal supports them. The attributes are ignored if the terminal does not support them.

On MVS/TSO, you can obtain a list of the available device support from the query reply structured field.

The IMS environments use the message format services to support terminal and printer maps. During generation, you can use the /MFSDEV, /MFSEATTR, and /MFSIGNORE options to specify device characteristics for all devices that use a map group. Refer to the *VisualAge Generator Generation Guide* for more details. Unpredictable results can occur if attributes are used that are not supported by the hardware. See “Considerations for Message Format Services in IMS” on page 105, for additional information concerning the message format services options.

Extended Addressing Considerations for VisualAge Generator Server

Some of the code provided with VisualAge Generator Server can run in extended addressing mode. This section describes considerations for using the extended addressing mode.

Most of the code shipped with VisualAge Generator Server runs in 31-bit addressing mode and resides above the 16MB line.

Most of the storage acquired by VisualAge Generator Server is above the 16MB line unless the first VisualAge Generator-generated program in the run unit is link-edited with AMODE(24) or is compiled with the /DATA=24 generation option. The AMODE(24) program attribute specifies that the program runs in 24-bit addressing mode. Refer to the *VisualAge Generator Generation Guide* for more information on AMODE(24) or the /DATA generation option.

Database Considerations

This section discusses preparing programs and checking access authorization to database resources when using the following:

- DB2 on MVS systems
- DB2/VSE on VSE systems and moving program packages among VSE systems
- SQL on VM systems

Preparing Programs

Before running a program, the SQL* statements need to be analyzed and prepared. You can choose to have your programs prepared by DB2 (the default on MVS systems), DB2/VSE (the default on VSE and VM systems), or an ANSI SQL database manager.

If you specify the /ANSISQL generation option, any SQL statements in the generated program are generated using the ANSI SQL format. If you use ANSI SQL, you need to tailor the templates, procedures, and execs for your database manager.

If you use DB2, you also need to bind the DB2 program plan. Refer to the [document](#) for information on binding DB2 program plans.

If your programs run in the MVS/TSO, MVS batch, or IMS BMP environments, you might also need to tailor the runtime JCL and CLIST templates. If your programs run in the VM CMS or VM batch environment, you might also need to tailor the runtime exec templates. Refer to the *VisualAge Generator Generation Guide* for additional information on tailoring runtime JCL, CLIST, and exec templates.

Checking Access Authorization

The database manager checks whether program users have the authority to access tables or run programs. The type of checking done varies depending on your system and the processing mode.

When accessing DB2 in generated COBOL programs, program users must be authorized to run the corresponding DB2 program plan.

DB2 requires an authorization identifier to ensure that program users have the DB2 authority to perform operations on the database and tables. The type of authorization checking done depends on whether the processing mode is static or dynamic. The authorization identifier of the program developer performing the BIND command is used for static SQL statements; the authorization identifier of the program user is used for dynamic SQL

statements. Generated COBOL programs use dynamic SQL statements for each process for which one of the following is true:

- Execution Time Statement Build option is specified for the statement
- The SQL table name associated with the SQL record is specified using a host variable

Any other SQL statements in the program are static statements. Refer to the DB2 administration manual for more information on the various ways the authorization identifier value is set.

Fold Tables

VisualAge Generator Server does folding (uppercase translation) using a 256-byte translate table. VisualAge Generator Server uses the translate table at offset X'0C' in CSECT ELANCccc where ccc is the language code.

Backing Up Data

You should regularly back up your data. This includes all files related to VisualAge Generator Server, private libraries, user-created data files, and user load libraries. System services are provided to back up and restore user libraries.

Customizing VisualAge Generator Server

Customizing VisualAge Generator Server consists of performing some of the same procedures used to install the product on the system. These procedures are described in the *Program Directory for VisualAge Generator Server* for your MVS, VSE or VM system. The program directory contains information on changing system options.

Chapter 8. System Considerations for MVS CICS

This chapter provides additional system requirements and considerations for administering VisualAge Generator Server in the MVS CICS environment.

The following information is discussed:

- Required file descriptions
- Segmented and nonsegmented processing
- Using transient data queues for printing
- Using the new copy function
- Recovery and database integrity considerations
- Considerations that affect performance
- Using and allocating data files
- Considerations for using DB2
- Considerations for using DL/I
- Considerations for using called remote programs

Required File Descriptions

VisualAge Generator Server requires the following files:

File Name

Description

ELAD This transient data queue is the default destination for VisualAge Generator Server error messages. VisualAge Generator Server produces error messages when it detects an error that prevents a program from continuing.

The ELAD transient data queue is defined when VisualAge Generator Server is installed. If you want to direct error messages for different transactions to different queues, define the other queues with the same characteristics as ELAD. Use the error diagnostic utility ELAC to direct error messages to the required queue. See the description of the utility in “Chapter 21. Diagnostic Control Options” on page 189 for more information.

EZEPRINT

The file that you associate to the VisualAge Generator file name EZEPRINT at resource association will be used when printing from a program that displays printer maps. This file can be defined with a

file type of SPOOL or TRANSIENT. This file is normally associated with the transient data queue EZEP.

If you installed VisualAge Generator Server as described in the VisualAge Generator Server program directory, EZEP is defined as an indirect destination associated with the system printer. The maximum record length that a generated program writes to the system printer is 650 bytes for double-byte character set (DBCS) print maps and 133 bytes for single-byte character set (SBCS) print maps. The first byte is an American National Standards printer control character. The DBCS record length is longer than the physical printer line length because the print record can contain outlining and shift-out/shift-in (SO/SI) control characters that do not appear on the device.

If you are using VisualAge Generator Server to print to a file destination other than EZEP, the characteristics of that file should be the same as EZEPRINT. The file destination can be specified either during resource association at generation time or with the EZEDESTP special function word at run time.

EZEPRMG

This VSAM indexed file (KSDS) contains the parameter group records used for print control options for the VisualAge Generator Server terminal printer utility, FZETPRT. The FZETPRT program reads this file searching for the parameter group matching the transaction name that started FZETPRT. See “Special Parameter Group for the FZETPRT Program” on page 59 for a description of the print parameters. See “Chapter 23. Using the Parameter Group Utility” on page 199 for more information about maintaining this special parameter group.

Segmented and Nonsegmented Processing

Generated programs can run in nonsegmented (CICS conversational) mode or segmented (CICS pseudoconversational) mode, and can switch between the two modes. When a program converses a map, a pseudoconversational terminal I/O operation is used if any of the following is true:

- The EZESEGM special function word is set to 1.
- The program issues an XFER statement with a map.
- The program was specified to run in segmented mode and the value was not overridden with the EZESEGM special function word.

The current transaction ends and the program status is saved in a temporary storage queue until the terminal input is received.

Two different storage queues are used. The storage queue names have the following format:

xyyytttt

where:

- x** Specifies a byte with the hex value X'EE'
- yyy** Specifies WRK (program working storage) or MSG (current map saved across help or error display)
- tttt** Specifies the terminal ID associated with the transaction

The /WORKDB generation option specifies whether a main or auxiliary storage queue is used. The storage queues are deleted at the end of a run unit.

Refer to the *VisualAge Generator Design Guide* for more information on developing segmented programs.

Using Transient Data Queues for Printing in MVS CICS

When a program displays a printer map, the printed output is sent to the logical file named EZEPRINT. The resource association file used by generation specifies whether the EZEPRINT file is a spool file or a transient data queue. Refer to the *VisualAge Generator Design Guide* for more information on using spool files for printing.

Processing a DISPLAY Process Option for a Printer Map

When printing is initiated through the DISPLAY process option for a printer map, the printer output is routed to the file that is specified as the resource associated to the EZEPRINT data set. The resource associated to EZEPRINT can be specified during resource association at generation time, or can be specified at run time using the EZEDESTP special function word. The remainder of this section describes how to set up transient data queues for printed output.

Printed output destined for a transient data queue is accumulated in temporary storage. The temporary storage queue name has the following format:

ttttnnnn

where:

- tttt** Is the transient data queue name
- nnnn** Is the EXEC Interface Block (EIB) task number

When a program ends, or a CLOSE process option for a print map is issued, or a segmentation break occurs, VisualAge Generator Server enqueues on a

transient data queue to prevent interspersed printing from other transactions. VisualAge Generator Server copies the printed output onto the transient data queue. The printed output is in line character format with an American National Standards printer-control character.

Specifying Print Destination

The default print destination in MVS CICS, VSE CICS, and VSE batch is a transient data queue named EZEP. If you installed VisualAge Generator Server as described in the VisualAge Generator Server program directory, EZEP is an indirect destination associated with the system printer. During program generation, this destination can be changed to any 4-character transient data queue name. The destination control table (DCT) entry for the queue determines the actual destination. The destination can be the system printer, a data set, or a terminal printer.

You can override the default destination at generation time by specifying the alternate destination as the system resource name for the EZEPRINT file. You can change the print destination at run time by using the EZEDESTP special function word. Refer to the VisualAge Generator online help system within VisualAge Generator Developer for additional information on the EZEDESTP special function word.

VisualAge Generator Server also provides a way of starting an asynchronous print task from a program and controlling the print destination from the program starting the asynchronous task. To do this, define the print task as a batch program and generate it with the option /PRINTDEST=TERMID. Use a CALL CREATX statement to start the batch program, specifying the print destination in the CREATX service routine parameters. The program ignores the generated print destination and uses the destination specified in the CREATX service routine. Refer to the VisualAge Generator online help system within VisualAge Generator Developer for more information on the CREATX service routine.

MVS CICS Terminal Printing

The program called FZETPRT supports terminal printing. This program runs as a CICS transaction that starts automatically when records are written to the transient data queue. If VisualAge Generator Server was installed as described in the VisualAge Generator Server program directory, the transaction name is EZEZ for IBM 5550-type printers and EZEP for all other printers. To send printed output to the terminal, you must include a TYPE=INTRA for the transient data queue in the CICS destination control table (DCT). Specify EZEP or EZEZ for the transaction ID in the DCT entry. The trigger level in that entry must be set to 1 to ensure proper output. See “Printing Transient Data at a Terminal Device” on page 72 for a sample DCT entry.

When the FZETPRT program is initiated, it reads a line from the transient data queue, converts the American National Standards printer-control character to NL EOM format, and writes to the terminal printer specified in the DCT entry. If you do not specify a terminal name in the DCT entry, the queue identifier must be the same as the terminal printer identifier. The FZETPRT program buffers multiple print lines into a single CICS SEND command to improve performance.

When using terminal printing with VisualAge Generator Server, you should be aware of potential problems regarding form-feed orders and page alignment. When the FZETPRT program is triggered, a form-feed order is issued to the printer to ensure that it begins printing at the top of a page. If a second map is sent to the queue before it is emptied by the FZETPRT program, a form-feed order is not issued before the second map is printed. Page alignment can vary depending on the timing with which successive maps are sent to the queue.

Another potential problem can occur when printing successive maps. If one of the maps in the series is defined with lines equal to, or one line fewer than, the lines-per-page setting on the printer, a blank page occurs between the printed maps. To avoid this, define the map size as 2 lines fewer than the lines-per-page setting on the printer. Because the FZETPRT program inserts a new-line order to ensure that printing begins in column 1, the first line of the map to be printed is actually printed on the second line of the page. The second line must be allowed because a new-line order is added after the last line of the map, which advances the print head to the beginning of the next line. If this happens to be the first line of the following page, the next form-feed order causes the page to be skipped before printing resumes.

Another thing to consider is that although VisualAge Generator Server sometimes causes successive, stand-alone form-feed orders (“1”), the FZETPRT program suppresses all but one of these in converting them to NL EOM format.

If these form-feed considerations are too restrictive for your needs, consider using the FORMFD=NO parameter described in “FORMFD Parameter” on page 62 .

Special Parameter Group for the FZETPRT Program

You can provide terminal printing parameters to the FZETPRT program to vary the printed output by using a special parameter group file.

The FZETPRT program attempts to read a file named EZEPRMG for a parameter group that has the same name as the transaction used to start the FZETPRT program. For example, if the print transaction that starts the

FZETPRT program is named EZEP, then FZETPRT tries to find the parameter group named EZEP. If the parameter group is not located in a file named EZEPRMG, or if EZEPRMG does not exist, then the FZETPRT program reads the DCAPRMG file to find the parameter group associated with this transaction.

When the transaction starts, the FZETPRT program reads the parameter group and varies the printer output according to the contents. If you need to use the terminal printing parameters, create a parameter group using the VisualAge Generator Server utility provided for this purpose. See “Chapter 23. Using the Parameter Group Utility” on page 199 for more information.

For this parameter group, you can specify the following four parameters:

- PRTBUF=xxx
- PRTMPP=nnn
- PRTTYP=D
- FORMFD=NO

Note: Do not include blanks between keywords and their associated values.

PRTBUF Parameter

Use the PRTBUF parameter to set the size of the printer buffer. The number of SEND commands sent to the terminal printer depends on the size of the printer buffer. The following example shows how to specify the buffer size using the PRTBUF parameter:

```
PRTBUF=xxx
```

where:

xxx Is the size in bytes of the printer buffer

The FZETPRT program uses a default buffer size if any of the following conditions occur:

- The parameter is not specified in the parameter group.
- There is no parameter group associated with the transaction.
- The parameter keyword is misspelled.
- The value specified is not valid (values greater than 8K bytes, smaller than 480 bytes, or not numeric).
- The EZEPRMG or DCAPRMG file does not exist or is not available.

The default buffer size is 2KB (where KB equals 1024 bytes) for the standard character set printers and 480 bytes for LU type 3 printers.

For double-byte character set (DBCS) users the default buffer size and the maximum buffer size allowed is 1918 bytes. The default value is used if your specified value exceeds the maximum number of bytes.

When the buffer size is larger than the default, usage of the PRTBUF parameter is optional. However, using the PRTBUF parameter is recommended to reduce the number of SEND commands sent to the terminal. If the printer buffer size is smaller than the default, specify the real buffer size using this parameter. Not specifying the real buffer size can cause unpredictable results.

PRTMPP Parameter

Use the PRTMPP parameter to set the maximum number of print positions. The following example shows how to specify the number of print positions using the PRTMPP parameter:

```
PRTMPP=nnn
```

where:

nnn Is the physical length (maximum print position) of the printer line

The FZETPRT program assumes a default maximum print positions of 132 if any of the following occurs:

- The parameter is not specified in the parameter group.
- There is no parameter group associated with the transaction.
- The parameter keyword is misspelled.
- The value specified is not valid (not numeric).
- The EZEPRMG or DCAPRMG file does not exist or is not available.

Use caution when coding the value of this parameter. If the value entered is a valid numeric, the FZETPRT program uses the value without validating it. If the value is greater than the number of print positions available on the actual printer, possible malfunctioning can take place causing more line skips than necessary.

Note: For DBCS users, this parameter must be specified unless the printer is configured with MPP=132.

PRTTYP Parameter

Use the PRTTYP parameter if you use a DBCS printer. The following example shows how to specify the use of a DBCS printer using the PRTTYP parameter:

```
PRTTYP=D
```

Note: This parameter must be used to specify that you are a DBCS user and your output is being directed to an IBM 5550-family printer.

If you use multiple printers with different characteristics (namely different MPP, different buffer size, or DBCS versus non-DBCS printers), you need as many transaction IDs as there are printers, each one associated with the FZETPRT program. For examples of table entries for two printers, see the CICS transaction definitions provided with VisualAge Generator Server for the EZEP (non-DBCS printers) and EZEZ (DBCS printers) transactions.

FORMFD Parameter

Use the FORMFD parameter to control the form-feed orders that the FZETPRT program issues. The following example shows the format of the FORMFD parameter:

```
FORMFD=NO
```

The FZETPRT program defaults to inserting form-feed orders into the printer data stream if any of the following occurs:

- The parameter is not specified in the parameter group.
- There is no parameter group associated with the transaction.
- The parameter does not appear as FORMFD=NO.
- The EZEPRMG or DCAPRMG file does not exist or is not available.

If the parameter is specified correctly, the FZETPRT program does not insert form-feed orders for any reason. This includes issuing a SET map PAGE statement, closing the printer, or the initial form feed that is normally done. All forms control depends on the map size specified during map definition.

CICS Entries for FZETPRT (DBCS only)

If you are using an SCS-type printer and you use DBCS, ensure that your system programmer has coded the destination control table (DCT) and the program control table (PCT) entries for a transaction that runs FZETPRT with the following option:

```
MSGPOPT=CCONTRL
```

The MSGPOPT option defines the optional facilities that a task can use. The CCONTRL parameter indicates that the program can control the outbound chaining of request units. Refer to the CICS manuals for more information.

Using the New Copy Function

The new copy function (either the VisualAge Generator Server new copy utility or the CICS NEWCOPY command) causes a transaction to use a new copy of a program, map group, or table referenced in the transaction. The VisualAge Generator Server new copy utility is implemented as a VisualAge Generator program in the CICS environment. Active transactions continue to use the current version of a program, map group, or table until the transaction either completes or reaches the end of a segment. A new copy of the program, map group, or table is then made available to the transaction by VisualAge Generator Server. Use the new copy function when programs, map groups, and tables are modified and generated again. This enables you to install new versions of programs, map groups, and tables onto your system without disrupting operation.

For programs and map groups, you can use the CICS NEWCOPY command or the VisualAge Generator Server new copy utility to cause the new copy of the program to be used the next time a load request is issued for the program. The VisualAge Generator Server new copy utility does a new copy for both the online print services program and the map group format module when you specify a part type of Map Group. If you use the CICS NEWCOPY command for a map group, you must issue the NEWCOPY for both the online print services program and the map group format module.

For tables, you must use the VisualAge Generator Server new copy utility to cause a fresh copy of the table to be used the next time a load request is issued for the table. Do not use the CICS NEWCOPY command for tables. The VisualAge Generator Server new copy utility sets a flag indicating that the new copy of the table is to be used the next time a program loads the table contents.

For more information on the VisualAge Generator Server new copy utility, see “New Copy” on page 183.

Recovery and Database Integrity Considerations

This section presents suggestions for specifying recovery options in CICS tables, and presents a discussion of the concept of a logical unit of work (LUW) that can help you determine your database recovery and integrity options.

VisualAge Generator programs can make use of all the MVS CICS recovery and data integrity features. Refer to the recovery and restart information for your release of CICS for a description of these features.

Specifying Recovery Options in the CICS Tables

The system initialization table (SIT) for CICS should specify DBP=XX, where XX is not equal to NO. If the DBP value is not equal to NO it prevents ASPE abends when generated programs issue CICS SYNCPOINT and CICS SYNCPOINT ROLLBACK commands.

If DTB=YES is specified on the program control table (PCT) entries for the transactions, the value specified for DBP is significant. CICS provides two dynamic backout programs, one for systems that require DL/I support and the other for systems that do not require DL/I support. These programs are provided by CICS if an entry is included in the processing program table (PPT) that specifies TYPE=GROUP and FN=BACKOUT.

Determining When Database Changes Are Committed

By understanding the concept of a logical unit of work (LUW) in VisualAge Generator programs, you are able to determine when changes are committed to your database. This section presents a definition of an LUW and some considerations for where to place commit points in your programs.

Using CICS recovery support, you can define your files and databases to be recoverable resources. If your program makes changes to a recoverable resource, the changes are not committed to the file or database until the end of an LUW. If your program ends abnormally before reaching the end of an LUW, all changes that were made since the beginning of the LUW are backed out.

An LUW for a transaction ends whenever a commit point or a rollback occurs. A commit point occurs when one of the following occurs:

- The run unit ends successfully. In MVS CICS, VSE CICS, and VSE batch, a run unit is equivalent to a single transaction and consists of all VisualAge Generator programs and non-VisualAge Generator programs that transfer control among themselves using a DXFR or CALL statement. For non-VisualAge Generator programs, this also includes any transfer that uses a CALL statement, CICS LINK command, or CICS XCTL command.
- A program uses a CONVERSE process option, and any of the following is set to 1:
 - EZESEGM special function word (segmented mode)
The EZESEGM special function word defaults to 1 if the program is defined as segmented.
 - EZECNVCM special function word (commit at every CONVERSE process option)
 - EZEDLTRM special function word (end the PSB at CONVERSE process option) if the program uses DL/I

These options cause the commit to be issued after terminal output and before the next terminal input. A commit point at terminal I/O synchronizes the updates to the database and confirmation messages to the program user.

- A transfer using an XFER statement occurs.
- A program calls either the EZECOMIT or COMMIT service routine.
- A transfer using a DXFR statement occurs and a PSB is scheduled, and one of the following occurs:
 - A transfer to a non-VisualAge Generator program and a PSB is scheduled
 - The /SYNCDXFR generation option was specified for the transferred-from program
 - The /NOSYNCDXFR generation option was specified for the transferred-from program and different PSB names were identified in the program specifications for the two programs
- A called DL/I program returns to the calling program, the PSB was not passed using the EZEDLPSB special function word, and PCBs were not passed using the EZEDLPCB special function word.

A rollback occurs when one of the following occurs:

- A program calls the EZEROLLB or RESET service routine.
- A program ends because of an error condition.

When a rollback occurs, all changes that were made to a database and recoverable files since the start of the LUW are backed out.

Considerations that Affect Performance

This section describes factors that affect system performance and suggestions on how to improve performance.

Residency (Modules in Memory) Considerations

The performance of a program is affected by is the number of times that a running program requires access to a disk. Programs require access to disks for the following reasons:

- Locating and loading VisualAge Generator Server load modules or phases
- Retrieving and storing user data
- Locating and loading application, map group format modules, online print services programs, and table load modules or phases

VisualAge Generator Server loads objects as they are needed. For example, VisualAge Generator Server loads an application, online print services

program, map group format module, or table when another program calls or references it. If you make an object resident, then the object remains in storage after it is loaded by VisualAge Generator Server. You can use the resident parameter on the PPT entry to make an application, online print services program, or map group format module resident.

For tables, the Shared, Resident, and Keep after use options are used to control residency. The Shared, Resident, and Keep after use options are definition attributes in VisualAge Generator Developer. Shared and Resident are specified within table definition and Keep after use is specified within program definition when specifying tables and additional records. These attributes cannot be overridden at generation time. If these attributes need to be changed, the definition must change. After generation, the change takes effect for all programs that contain the respective tables.

Virtual Storage Considerations and Residency

It is true that if an application, online print services program, map group format module, or table is resident, less I/O is required for multiple loads. However, making these objects resident requires more virtual storage because the modules accumulate in storage as they are loaded and are not deleted after they are used.

When deciding what to make resident, consider the following:

- Storage constraints
- Frequency of program use
- Long running programs versus programs that are started more frequently

Because most systems have virtual storage constraints, it is not possible to make everything resident. You should establish priorities for deciding which objects you want to make resident. These residency priorities reflect a trade-off between program usage and storage constraints. Your priorities can dictate that some components of a program (such as the online print services program and the map group format module) must be made resident, while other components (such as tables) must not be made resident.

Generally, objects that are loaded more than once are prime candidates for residency. Examples of this are a table that is used by more than one program or a program that is called more than once.

The following are some general rules for residency:

- When deciding what to make resident, consider the following objects:
 - Called programs
 - Online print services program or module

- Map group format modules
- Tables
- Main programs
- Programs that are started or referenced frequently should be made resident. This includes programs that are started by VisualAge Generator CALL or DXFR statements.
- When defining tables, you should specify Keep after use within program definition when specifying tables and additional records to prevent VisualAge Generator Server from deleting the table after it is used. In addition, if you want to make a table resident, use Resident and Shared within table definition.
- Programs that are not frequently initiated or have long running time should not be made resident.

In MVS CICS or VSE CICS, when a program component is made resident, it remains in storage from the time it is loaded into storage until either CICS is shut down or the new copy function is used. To aid in deciding which programs should be made resident, you can use CICS shutdown statistics to determine how often a generated program or other component is loaded into the region or partition.

Pseudoconversational Programs and Residency

In MVS CICS or VSE CICS, if you plan to run a program in pseudoconversational mode, you should consider making all components of the program resident. In pseudoconversational mode, the program and its components are deleted and loaded again at each segment break (for example, at each segmented CONVERSE process option) if they are not made resident. This degrades the performance of pseudoconversational mode.

Work Database Temporary Storage Queue Considerations

When running in pseudoconversational mode (using a segmented CONVERSE process option), the data and the status associated with the program must be saved during user think time. You can control whether this information is saved into the CICS main temporary storage or auxiliary storage. Using main temporary storage can result in better performance because the data is written to memory within the CICS address space instead of writing the data to disk space.

The /WORKDB generation option can be used to choose whether main or auxiliary temporary storage is used for the work database. See the *VisualAge Generator Generation Guide* for more information about specifying this option.

Note: Use of main temporary storage can degrade system performance because the increased address space that is referenced can increase the paging activity. Also, CICS can experience a short-on-storage condition if the program data to be saved exceeds the available CICS storage. Therefore, if you take advantage of main temporary storage for programs requiring better performance, you should monitor your system to ensure that virtual storage problems do not occur.

The amount of data written or read on each request to CICS when saving program data and status, can also affect performance. The installation options module, ELARPIOP, specifies the largest size record VisualAge Generator Server writes to main or auxiliary temporary storage. The default size is 32KB (where KB equals 1024 bytes), which is the largest value allowed by CICS. Use a large value to ensure that the least number of write requests are required, and, if using auxiliary storage, to ensure that the least number of I/O operations are required. See the *VisualAge Generator Server Program Directory* for information on how to change the value in the installation options module.

Note: If you are using auxiliary storage queues, you should ensure the control interval size (CISIZE) of the VSAM data set used for auxiliary temporary storage matches the size specified in the installation options file. If the CISIZE for the data set is smaller, CICS splits the data written or read into smaller pieces and does multiple I/O operations for each VisualAge Generator Server request. Also ensure that you have an adequate number of buffers for the auxiliary temporary storage data set in order to reduce the number of physical I/O operations.

Database Temporary Storage Size

The installation options module, ELACPIOP, specifies the largest size record that can be written to main or auxiliary temporary storage when saving program status on a segmented CONVERSE process option. The size specified must be less than or equal to the control interval (CI) size defined for the VSAM data sets used for auxiliary storage queues at your installation. Using a larger CI size might improve performance. However, the maximum CI size that can be specified is 32KB (where KB equals 1024 bytes). To change the value in the installation options module, see the *Program Directory for VisualAge Generator Server for MVS*.

Conversational and Pseudoconversational Processing

Conversational processing can be more efficient than pseudoconversational. If storage and other CICS resources are not a constraint on an XA or ESA system, define programs as nonsegmented, and set the EZECNVCM special function word in the program to 1. Setting the EZECNVCM special function word to 1 causes file and database resources to be released for each

CONVERSE process option. This combines the better performance of nonsegmented processing with the resource-sharing characteristics of segmented processing.

Terminal Printing

The performance of terminal printing can be enhanced by specifying the PRTBUF parameter for the FZETPRT program. See “MVS CICS Terminal Printing” on page 58 for more information on terminal printing and the PRTBUF parameter.

Monitoring and Tuning the System

For MVS CICS, you can use the CICS MVS Performance Analysis Reporting System (CICSPARS/MVS) to monitor performance. This tool provides information about current tasks, snapshots of the current system status, and information about the MVS CICS dynamic storage area, as well as the ability to display MVS CICS storage.

For CICS/ESA, you can use the CICS/ESA monitoring facility to get information about CICS/ESA tasks.

Refer to the performance guide for your release of MVS CICS for more information.

Using and Allocating Data Files in MVS CICS and VSE CICS

This section describes how to define data files for use in generated VisualAge Generator programs in the MVS CICS or VSE CICS environment.

Defining and Loading VSAM Data Files

Before CICS programs can use VSAM data files, you must define and load them. See “Defining and Loading VSAM Program Data Files” on page 47 for information on defining VSAM data sets, defining an alternate index, and loading a VSAM data set.

Adding the Job Control Statements

After the data set has been defined and loaded, add the data set name to the CICS startup JCL or the VSE/ESA standard label area to allocate user files. You can also let MVS CICS dynamically allocate the data set to the file using the information specified in the file control table (FCT). Figure 5 on page 70 shows example allocation statements for an indexed, relative, and serial file, and an alternate index.

```
//KSDSFILE DD DSN=ELA1.USER.KSDS,DISP=SHR
//RRDSFILE DD DSN=ELA1.USER.RRDS,DISP=SHR
//ESDSFILE DD DSN=ELA1.USER.ESDS,DISP=SHR
//KSDSAIX DD DSN=VSAM.KSDS.ALT.INDEX.PATH,DISP=SHR
```

Figure 5. Allocating User Files

Adding the File Name to the CICS File Control Table

After the data set has been defined, loaded, and added to the CICS startup JCL, the FCT entry must be created for the file name for a CICS program to gain access to the data set. Creating an FCT entry can be accomplished using online (RDO) or macro definitions.

Figure 6 on page 71 shows sample macro definition entries that can be used to add a file name to the FCT. VisualAge Generator Server uses the name on the FILE operand. The FILE operand name must be the same as the ddname (MVS) or the DLBL name (VSE) in the CICS startup JCL. All other operands must be the same as when you add an indexed, relative, or serial file to the FCT.

With MVS CICS or VSE CICS, make an entry to the FCT for every file used by a program. The CICS files can be defined as remote FCT entries.

For further information, refer to the appropriate CICS resource definition guide for your environment.


```

*** THIS IS A SAMPLE OF AN FCT ENTRY FOR A KEY-SEQUENCED ****
*** DATA SET(KSDS). ****
    DFHFCT TYPE=FILE, C
        FILE=KSDSFILE, C
        ACCMETH=VSAM, C
        SERVREQ=(READ,UPDATE,ADD,DELETE,BROWSE), C
        FILSTAT=(ENABLED,CLOSED), C
        RECFORM=FIXED
*** END OF KSDS *****
*** THIS IS A SAMPLE OF AN FCT ENTRY FOR AN ALTERNATE INDEX **
*** DATA SET(KSDS). ****
    DFHFCT TYPE=FILE, C
        FILE=KSDSAIX, C
        ACCMETH=VSAM, C
        BASE=KSDS, C
        SERVREQ=(READ,UPDATE,ADD,DELETE,BROWSE), C
        FILSTAT=(ENABLED,CLOSED), C
        RECFORM=FIXED
*** END OF KSDS *****
*** THIS IS A SAMPLE OF AN FCT ENTRY FOR A RELATIVE RECORD ***
*** DATA SET(RRDS). ****
    DFHFCT TYPE=FILE, C
        FILE=RRDSFILE, C
        ACCMETH=VSAM, C
        SERVREQ=(READ,UPDATE,ADD,DELETE,BROWSE), C
        FILSTAT=(ENABLED,CLOSED), C
        RECFORM=FIXED
*** END OF RRDS *****
*** THIS IS A SAMPLE OF AN FCT ENTRY FOR AN ENTRY-SEQUENCED ***
*** DATA SET(ESDS). ****
    DFHFCT TYPE=FILE, C
        FILE=ESDSFILE, C
        ACCMETH=VSAM, C
        SERVREQ=(READ,UPDATE,ADD,DELETE,BROWSE), C
        FILSTAT=(ENABLED,CLOSED), C
        RECFORM=FIXED
*** END OF ESDS *****
***** E N D   O F   U S E R   D A T A   F I L E S *****

```

Figure 6. Adding the File Name to the FCT

Using Remote Files

VisualAge Generator programs can gain access to files that do not reside on your MVS CICS or VSE CICS system. Refer to the *VisualAge Generator Client/Server Communications Guide* for more information. Refer to the section on developing client/server programs in the *VisualAge Generator Generation Guide* for information on gaining access to remote files from generated programs on either a workstation or a CICS host session. Also refer to the *VisualAge Generator Generation Guide* for information on specifying linkage tables. Refer to the appropriate CICS manuals for information about defining remote programs, transactions, or files.

Defining Transient Data Queues

Transient data queues are used in MVS CICS and VSE CICS for reading or writing data from tapes, disks, or other sequential (MVS) or serial (VSE) files. If you associated a serial or print file with a transient data queue at generation, you must define a MVS CICS or VSE CICS destination control table (DCT) entry for the queue.

You can define the following types of transient data queues:

- Intrapartition (temporary data)
- Extrapartition (data that other non-CICS regions can use)

Intrapartition transient data files contain data that is not usable after it is read.

Defining Intrapartition Transient Data

The following two examples show how to define intrapartition transient data files.

Passing Transient Data between CICS Transactions: This is an example of a DCT entry that can be used to pass data from one CICS transaction to another. The file destination specified at generation with the SYSNAME option should be xxxx.

```
DFHDCT TYPE=INTRA,           C
      DESTID=xxxx,           C
      DESTFAC=FILE
```

Printing Transient Data at a Terminal Device: This is an example of a DCT entry that can be used for terminal printing in VisualAge Generator Server. At generation, the print destination for EZEPRINT can be specified. The default is EZEP. (A DCT entry is supplied for EZEP that sends the printed output to the system printer.) The program supplied for printing, FZETPRT, reads records from the transient data queue and issues SEND commands to the terminal in order to print the records.

In this sample DCT, the PR01 terminal is to receive the printed output. PR01 is a MVS CICS or VSE CICS printer terminal name. You specify the EZEPRINT destination at generation as PR01. VisualAge Generator Server writes the printed output to the transient data queue, PR01. The transaction EZEP starts and causes the program FZETPRT to run. The data is read from the transient data queue and sent to the terminal, PR01. The program control table (PCT) entries for EZEP and the processing program table (PPT) entries for FZETPRT are supplied. You must supply the destination control table and the terminal control table entries for the transient data and terminal.

```

DFHDCT TYPE=INTRA,           C
      DESTID=PR01,           C
      DESTFAC=TERMINAL,      C
      TRANSID=EZEP,          C
      TRIGLEV=1

```

If the terminal printer is a DBCS printer, specify EZEZ as the TRANSID.

Defining Extrapartition Transient Data

Data to be read from tape or sent to a printer is contained in extrapartition transient data queues.

The following two examples show how to use extrapartition transient data queues. These files can be used by non-CICS devices and by CICS.

Printing Transient Data: This is an example of a DCT entry specification that can be used to print output on a high-speed system printer. The file destination specified at generation with the SYSNAME option should be *zzzz*.

You need to add the appropriate JCL to the CICS runtime JCL to assign a printer to the file name. The following sample entry for the DCT is for printed output.

```

DFHDCT TYPE=EXTRA,           C
      DESTID=ZZZZ,           C
      DSCNAME=PRINTER
DFHDCT TYPE=SDSCI,           C
      DSCNAME=PRINTER,       C
      RECFORM=VARBLKA,        C
      RECSIZE=133,            C
      BLKSIZE=1330,           C
      TYPEFLE=OUTPUT          C

```

The JCL used in the extrapartition destination data queue sample requires the following JCL:

```
//PRINT DD SYSOUT=*,DCB=(RECFM=VBA,LRECL=133,BLKSIZE=1330)
```

Considerations for Using DB2 in MVS CICS

This section presents considerations for programs that access DB2 databases, and recovery and database integrity for DB2 programs running in the MVS CICS environment.

For information on designing and generating DB2 programs for the MVS CICS environment, refer to the *VisualAge Generator Design Guide* or the *VisualAge Generator Generation Guide*.

For information on preparing DB2 programs for running on the MVS CICS environment, see Chapter 15. Preparing and Running a Generated Program in MVS CICS.

Associating DB2 Databases with CICS Transactions

If the programs running under a transaction access DB2 databases, then you must define an entry in the CICS resource control table (RCT) that associates the transaction identifier with the program plan name. The following example shows the minimum RCT entry required:

For more information on the other parameters you can specify when you

```
DSNCRCT TYPE=ENTRY TXID=tran PLAN=plan-name
```

define RCT entries, refer to the chapter on connecting the CICS attachment facility in the DB2 installation manual for your version of DB2.

Recovery and Database Integrity Considerations

VisualAge Generator programs can use all the recovery and data integrity features that are provided by DB2 in the MVS CICS environment or SQL in the VSE CICS environment.

Relational databases are recoverable resources. If your program makes changes to a relational database, the changes are not committed to the database until the end of a logical unit of work (LUW). If your program ends abnormally before the end of an LUW, all changes that were made since the beginning of the LUW are backed out. See “Recovery and Database Integrity Considerations” on page 63 for additional information.

Considerations for Using DL/I in MVS CICS

This section discusses recovery and database integrity considerations for DL/I programs running in the MVS CICS environment.

For information on designing and generating DL/I programs for the MVS CICS environment, refer to the *VisualAge Generator Design Guide* or the *VisualAge Generator Generation Guide*.

For information on preparing DL/I programs for running in the MVS CICS environment, see Chapter 15. Preparing and Running a Generated Program in MVS CICS.

Recovery and Database Integrity Considerations

VisualAge Generator programs can make use of all the recovery and data integrity features that are provided by DL/I in the MVS CICS and VSE CICS environments.

DL/I databases are recoverable resources. If your program makes changes to a DL/I database, the changes are not committed to the database until the end of a logical unit of work (LUW). If your program ends abnormally before the end of an LUW, all changes that were made since the beginning of the LUW are backed out. See “Recovery and Database Integrity Considerations” on page 63 for additional information.

Chapter 9. System Considerations for MVS/TSO

This chapter provides additional administrative information that applies to the MVS/TSO environment.

The following information is discussed:

- Required File Descriptions
- Printing Considerations
- Making VisualAge Generator Server, COBOL, and the Generated Modules Available
- Recovery Considerations
- Using VSAM Program Data Files
- Considerations for Using DB2
- Considerations for Using DL/I

Required File Descriptions

VisualAge Generator Server requires the following files:

File Name	Description
EZEPRINT	<p>This file is used when printing from a program that displays printer maps. The sample CLIST is set up to print to a disk data set; however, you can customize the CLIST to print to another destination. EZEPRINT can be allocated to either a data set or to a SYSOUT class. The file must have a VBA (variable-blocked ANSI) record format.</p> <p>The maximum record length that a generation program can write to the print data set is 654 bytes for DBCS maps and 137 bytes for SBCS maps. The record length includes 4 bytes for the variable length record header, 1 byte for the American National Standards printer-control character, and the print line for the print map. The DBCS record length is longer than the printer line length because the print line can contain outlining control characters and shift-out (SO) and shift-in (SI) characters that are not displayed on the device. The logical record length defined for the data set must be greater than or equal to the length of the longest line written by the program, including the DBCS SO/SI characters.</p> <p>If you are using VisualAge Generator Server to print to a file destination other than EZEPRINT, the characteristics of that</p>

file should be the same as EZEPRINT. The file destination can be specified either during resource association at generation time, or at run time with EZEDESTP special function word.

SYSPRINT, SYSOUT, SYSABOUT, SYSUDUMP

These VisualAge Generator Server system output files are used by generated programs. Do not specify DCB parameters for these files.

ELAPRINT This system output file is used by generated programs. Specify ELAPRINT with RECFM=FBA and BLKSIZE=1330 DCB parameters.

ELATRACE This file is the trace control file for the MVS/TSO environment. The attributes for this data set are LRECL=80, RECFM=FB, and BLKSIZE=multiple of 80. The trace filters are specified in the ELATRACE data set.

ELATOUT The output of the VisualAge Generator Server trace facility is sent to this data set in the MVS/TSO environment. The attributes for this data set are DSORG=PS, LRECL=133, BLKSIZE=1330, and RECFM=FBA.

Printing Considerations for MVS/TSO

Generated program printing is routed to the file that is specified as the resource associated to EZEPRINT. If a file destination other than EZEPRINT is specified either during generation or at run time (using the EZEDESTP special function word), output is directed to the specified destination. For example, if the program developer generated the program with EZEPRINT specified as ALTPRINT, you must allocate the file name ALTPRINT and direct the print output to it before running the program. If you do not specify the proper file name, the print connection is not successful. Refer to the online help system for more information on the EZEPRINT and EZEDESTP special function words.

Terminal printing is not directly supported in MVS/TSO.

Making VisualAge Generator Server, COBOL, and the Generated Modules Available

If your generated program contains DL/I or SQL processes, use one of the following methods to obtain access to the load library where your generated programs reside:

- Add the library to the STEPLIB concatenation list for your MVS/TSO logon procedure.

- Use the ISPF LIBDEF ISPLLIB command.
- Use the TSOLIB command provided by TSO.

The MVS/TSO CALL command that is used to start the program establishes a temporary task library (TASKLIB). This task library is searched when locating or loading the load module. The task library is only available during the CALL command.

Note: The preceding methods are not required if neither DL/I nor DB2 are used and all the programs (applications, online print services programs, map group format modules, table programs, and non-VisualAge Generator programs) for the run unit are in the same load library.

For all types of programs, the VisualAge Generator Server and the LE runtime libraries must be made available using one of the preceding methods. These libraries might have been included in the STEPLIB concatenation list when VisualAge Generator Server was installed and customized. Your system programmer might have placed these runtime libraries in the LNKLIST concatenation list, making them available for all users of the system.

Recovery Considerations

A logical unit of work (LUW) for an MVS/TSO program ends whenever a commit point or a rollback occurs.

A commit point occurs in MVS/TSO when one of the following occurs:

- The top-level program in a run unit ends successfully and returns control to the MVS/TSO command processor.

For MVS/TSO, a run unit consists of all VisualAge Generator programs and non-VisualAge Generator programs that transfer control among themselves using an XFER, DXFR, or CALL statement. For non-VisualAge Generator programs, this also includes any transfer that uses an OS XCTL command or an OS CALL statement.

- A program uses a CONVERSE process option, and any of the following is set to 1:
 - EZESEGM special function word (segmented mode)
The EZESEGM special function word defaults to 1 if the program is defined as segmented.
 - EZECONVCM special function word (CONVERSE commit)
 - EZEDLTRM special function word (end the PSB at CONVERSE) if the program uses DL/I

The best time for a commit point to occur is after terminal output and before the next terminal input. A commit point at terminal I/O synchronizes updates to the database and confirmation messages to the program user.

- A transfer using an XFER statement occurs for a segmented or single segment program.
- A program transfer using an XFER statement and the /SYNCXFER generation option is specified for the transferred-from program.
- A program calls the EZECOMIT service routine.

For MVS/TSO, VisualAge Generator programs that do not use DL/I issue a commit point only if the program has made changes to an SQL table. A commit point does not occur for changes to an SQL table made by a non-VisualAge Generator program.

A rollback occurs in MVS/TSO when one of the following occurs:

- A program calls the EZEROLLB service routine.
- A program ends because of an error condition.

When a rollback occurs, all changes that were made to databases since the start of the LUW are backed out.

Using VSAM Program Data Files in MVS/TSO

VSAM program data files must be defined before your MVS/TSO program can use them. Use the IDCAMS program to define a user VSAM data file to MVS/TSO. See “Defining and Loading VSAM Program Data Files” on page 47 for information on defining VSAM data sets, defining alternate indexes, and for information on loading VSAM data sets.

You can either allocate user data sets by using the MVS/TSO ALLOCATE command or let VisualAge Generator Server allocate them dynamically. Refer to the *VisualAge Generator Design Guide* for more information on file allocation.

Considerations for Using DB2 in MVS/TSO

This section presents system considerations for database recovery and integrity for DB2 programs.

For information on designing, generating, and preparing DB2 programs for the MVS/TSO environment, refer to the *VisualAge Generator Design Guide* and the *VisualAge Generator Generation Guide*.

For information on running DB2 programs in MVS/TSO, see Chapter 16. Preparing and Running a Generated Program in MVS/TSO.

Recovery and Database Integrity Considerations

VisualAge Generator programs can use all the recovery and data integrity features provided by DB2.

Relational databases are recoverable resources. If your program makes changes to a relational database, the changes are not committed to the database until the end of a logical unit of work (LUW). If your program ends abnormally before the end of an LUW, all changes that were made since the beginning of the LUW are backed out. See “Recovery Considerations” on page 79 for information describing when an LUW ends.

Considerations for Using DL/I in MVS/TSO

This section presents the following information:

- DL/I support in generated programs
- Defining the program specification block
- Recovery and database integrity considerations

For information on designing, generating, and preparing DL/I programs for the MVS/TSO environment, refer to the *VisualAge Generator Design Guide* and the *VisualAge Generator Generation Guide*.

For information on running DL/I programs in the MVS/TSO environment, see “Chapter 16. Preparing and Running a Generated Program in MVS/TSO” on page 151.

DL/I Support in Generated Programs

Programs generated for the MVS/TSO environment can gain access to either DL/I or DB2 databases, but not both. The command list (CLIST) used to start the program starts the program using the DL/I batch interface. You can customize the CLIST to use the IMS BMP interface instead.

Defining the Program Specification Block (PSB)

The following lists the considerations for using DL/I PSBs on MVS/TSO:

- DL/I PSBs used in the MVS/TSO environment must have `CMPAT=YES` specified in the `PSBGEN` statement for the PSB. This enables you to use the `CHKP` and `ROLB` functions with the PSB.

- The PSBGEN statement must include the parameter LANG=COBOL or LANG=ASSEM.
- DL/I PSBs used in the MVS/TSO environment must be defined with a minimum of two PCBs of any type in the PSB. This enables the generated COBOL program to test whether it is being started from the IMS region controller or by an OS XCTL macro in a non-VisualAge Generator program passing working storage and the EZEDLPSB special function word as parameters.

Recovery and Database Integrity Considerations

In MVS/TSO DL/I programs, a commit point causes a DL/I basic CHKP (checkpoint) call. The contents of the EZEDLPSB special function word are used as the checkpoint identifier. After the CHKP call, the EZEDLSTC special function word contains the status code returned with the CHKP call.

In MVS/TSO DL/I programs, a rollback causes a DL/I ROLB (rollback) call. After the ROLB call, the EZEDLSTC special function word contains the status code returned with the ROLB call.

See “Recovery Considerations” on page 79 for information describing when a commit or rollback occurs.

Chapter 10. System Considerations for MVS Batch

This chapter presents system considerations for running VisualAge Generator-generated programs in the MVS batch environment.

The following information is discussed:

- Required file descriptions
- Recovery considerations
- Using VSAM program data files
- Considerations for using DB2
- Considerations for using DL/I
- Runtime JCL

Required File Descriptions

Programs generated for the MVS batch environment use the same files as those generated for the MVS/TSO environment. See “Required File Descriptions” on page 77 for information on the required files for MVS batch.

Recovery Considerations

A logical unit of work (LUW) for an MVS batch program ends whenever a commit point or a rollback occurs.

A commit point occurs in MVS batch when one of the following occurs:

- The top-level program in a run unit ends successfully.
For MVS batch, a run unit consists of all VisualAge Generator programs and non-VisualAge Generator programs that transfer control among themselves using an XFER, DXFR, or CALL statement. For non-VisualAge Generator programs, this also includes any transfer that uses an OS XCTL macro or CALL statement.
- A program transfer using an XFER statement and the /SYNCXFER generation option is specified for the transferred-from program.
- A program calls the EZECOMIT or COMMIT service routine.
For MVS batch, VisualAge Generator programs that do not use DL/I issue a commit point only if the program has made changes to an SQL table. A commit point does not occur for changes to an SQL table made by a non-VisualAge Generator program.

A rollback occurs in MVS batch when one of the following occurs:

- A program calls the EZEROLLB or RESET service.
For MVS batch, VisualAge Generator programs that do not use DL/I issue a rollback only if the program has made changes to an SQL table. A rollback does not occur for changes to an SQL table made by a non-VisualAge Generator program.
- A program ends because of an error condition.

When a rollback occurs, all changes that were made to databases since the start of the LUW are backed out.

Using VSAM Program Data Files in MVS Batch

VSAM program data files must be defined before your MVS batch program can use them. See “Defining and Loading VSAM Program Data Files” on page 47 for information on defining VSAM data sets, defining alternate indexes, and for information on loading VSAM data sets.

The DD statements for user files are generated for you and placed in the sample runtime JCL.

Considerations for Using DB2 in MVS Batch

This section presents system considerations for database recovery and integrity for DB2 programs.

For information on designing, generating, and preparing DB2 programs for the MVS batch environment, refer to the *VisualAge Generator Design Guide* and the *VisualAge Generator Generation Guide*.

For information on running DB2 programs in MVS batch, see “Chapter 17. Preparing and Running Generated Programs in MVS Batch” on page 159.

Recovery and Database Integrity Considerations

VisualAge Generator programs can use all the recovery and data integrity features provided by DB2.

Relational databases are recoverable resources. If your program makes changes to a relational database, the changes are not committed to the database until the end of an LUW. If your program ends abnormally before the end of an LUW, all changes that were made since the beginning of the

LUW are backed out. See “Recovery Considerations” on page 83 for information describing when an LUW ends.

Considerations for Using DL/I in MVS Batch

This section presents the following information:

- Defining the program specification block (PSB)
- Recovery and database integrity considerations

For information on designing, generating, and preparing DL/I programs for the MVS batch environment, refer to the *VisualAge Generator Design Guide* and the *VisualAge Generator Generation Guide*.

For information on running DL/I programs in the MVS batch environment, see “Chapter 17. Preparing and Running Generated Programs in MVS Batch” on page 159.

Defining the Program Specification Block (PSB)

The following list shows considerations for defining a PSB that is used in the MVS batch environment:

- DL/I PSBs used in the MVS batch environment must have CMPAT=YES specified in the PSBGEN statement for the PSB. This enables you to use the CHKP and ROLB functions with the PSB.
- The PSBGEN statement must include the parameter LANG=COBOL or LANG=ASSEM.
- DL/I PSBs used in the MVS batch environment must be defined with a minimum of two PCBs of any type in the PSB. This enables the generated COBOL program to test whether it is being started from the IMS region controller or from an OS XCTL macro in a non-VisualAge Generator program passing working storage and the EZEDLPSB special function word as parameters.
- MVS batch programs can implement serial files as GSAM databases. These GSAM files are treated as a special type of database and require a PCB in the PSB. The GSAM PCBs must follow all database PCBs.

Recovery and Database Integrity Considerations

In MVS batch DL/I programs, a commit point causes a DL/I basic CHKP (checkpoint) call. The contents of the EZEDLPSB special function word are used as the checkpoint identifier. After the CHKP call, the EZEDLSTC special function word contains the status code returned with the CHKP call.

If the program runs under the MVS/TSO terminal monitor program for SQL access, calling the EZEROLLB service routine results in an SQL ROLLBACK WORK.

If the program runs as a DL/I batch job, and DL/I or SQL requests have been issued, calling the EZEROLLB service routine results in a DL/I ROLB call. The IMS batch parameter BKO=Y must be specified when the batch job is started in order for the ROLB call to be honored. The BKO parameter is specified in the job step that calls the IMS control program DFSRRC00. If BKO=N is specified, DL/I returns status code AL for the ROLB call. VisualAge Generator Server treats the AL as a soft error, and no error message is issued.

Serial or print files associated with GSAM files and AUDIT service routine calls result in DL/I requests and cause the DL/I ROLB call to be issued. See “Recovery Considerations” on page 83 for information describing when a commit or rollback occurs.

Performance Considerations for MVS Batch

See “Modules in Memory” on page 46 for information on performance considerations and the methods used to place modules in memory. These methods are particularly beneficial if the VisualAge Generator program is being called repeatedly by a non-VisualAge Generator program.

If you are running generated programs in MVS batch, you no longer need to perform an UPDATE process option on the record prior to a DELETE or REPLACE process option. Eliminating the UPDATE option allows for better performance, as it eliminates the COBOL read performed by the UPDATE process option. However, make sure that a previous SCAN or INQUIRY has been done before performing a DELETE or REPLACE without a preceding UPDATE. A SCAN or INQUIRY is adequate for ensuring that the record has been read and is ready for the DELETE or REPLACE.

Runtime JCL

See “Chapter 17. Preparing and Running Generated Programs in MVS Batch” on page 159 for examples of batch runtime JCL.

Chapter 11. System Considerations for IMS

This chapter provides additional administrative information that applies to the IMS environments.

The following information is discussed:

- Required file descriptions
- Defining the program specification block
- Processing modes
- Printing considerations for IMS
- Recovery and database integrity considerations
- Considerations that affect performance
- Considerations for using DB2
- Considerations for using DL/I
- Maintaining the work database
- Considerations for message format services

Required File Descriptions

VisualAge Generator Server requires the following files:

File Name	Description
ELASNAP	This is an optional file that contains the snap dump listing when a VisualAge Generator Server error occurs and the ELASNAP DD statement was included in the startup JCL. This file has a 125-byte logical record length, a 882-record block size, and a VBA (variable-blocked ANSI) record format. If this file is directed to the system logical unit SYSOUT define it with RECFM=VBA and BLKSIZE=4096.
ELAPRINT	This file is an optional output file for VisualAge Generator Server error messages. This file has a fixed block record format, a 133-byte logical record length, and a block size of 1330. If this file is directed to the system logical unit SYSOUT, define it with RECFM=FBA BLKSIZE=1330.
ELADIAG	This is the default name for the optional message queue for VisualAge Generator Server error messages. This message queue is defined in the IMS system definition during VisualAge Generator Server installation. Refer to

“Chapter 24. IMS Diagnostic Message Print Utility” on page 203 for information about printing the error messages contained in the ELADIAG message queue.

ELATRACE	This is the trace control file for the IMS BMP environment. The attributes for this data set are LRECL=80, DSORG=PS, and BLKSIZE=multiple of 80. The trace filters are specified in the ELATRACE data set.
ELATOUT	The output of the VisualAge Generator Server trace facility is sent to this data set in the IMS BMP environment. The attributes for this data set are LRECL=133, BLKSIZE=1330, and RECFM=FBA.
ELAT	The output of the VisualAge Generator Server trace facility is sent to this output message queue in the IMS/VS environment. Use the ELAMQJUD job to retrieve the trace.
EZEPRINT	This is the default message queue (IMS/VS) or output file (IMS BMP) for print output from generated programs. For IMS BMP programs, the print records are variable length. For single-byte languages, define EZEPRINT with LRECL=137, BLKSIZE=141, and RECFM=VBA. For double-byte languages, define EZEPRINT with LRECL=654, BLKSIZE=658, and RECFM=VBA. If the file is directed to the system logical unit SYSOUT, define it with RECFM=VBA, BLKSIZE=4096.

Defining the Program Specification Block (PSB)

You need to define both an IMS PSB and a VisualAge Generator PSB for your program. The VisualAge Generator PSB contains a subset of the information from the IMS PSB and is used to build default segment search arguments (SSAs) for the VisualAge Generator process options.

You need to generate an IMS PSB to correspond to the VisualAge Generator PSB. For IMS/VS, the IMS PSB must have the same name as the load module for the associated COBOL program. A program control block (ACB) generation is also required for the IMS/VS environment. For BMP and DL/I batch, the IMS PSB name does not have to match the program load module name.

When you define the PSBs for IMS programs, consider the following criteria:

- The I/O PCB (program control block) is automatically supplied and does not appear in the IMS or VisualAge Generator PSB source.

- Alternate PCBs are used to route output to terminals other than the originating terminal, or to other transactions. Alternate PCBs must appear before the database PCBs both in the IMS and the VisualAge Generator PSB source.
- When a VisualAge Generator program is generated for the IMS/VS or IMS BMP environment, a modifiable alternate PCB and a modifiable express alternate PCB are required, in that order, as the first two PCBs following the I/O PCB. Both of these PCBs must have the parameters ALTRESP=NO and SAMETRM=NO. To avoid having to edit your DL/I call modifications to adjust for the two required PCBs, include these PCBs whenever you plan to generate a program for the IMS/VS or IMS BMP target environments. The VisualAge Generator test facility handles their presence in the VisualAge Generator PSB.
- The PSBGEN statement must include the parameters CMPAT=YES, and LANG=COBOL or LANG=ASSEM.
- IMS BMP programs can implement serial files as GSAM databases. These GSAM files are treated as a special type of database and require a PCB in the PSB. The GSAM PCBs must follow all database PCBs.

If a DL/I work database is used, the PCB for this database must be included in the IMS PSB. This PCB can be created using the macro ELAPCB and concatenating ELA.V1R2M0.ELASAMP as part of the SYSLIB in the PSBGEN procedure. Figure 7 shows an example of the PCB expansion that occurs when ELAPCB is used.

WORKDBD defaults to ELAWORK. The WORKDBD parameter must be used if the DBD name is changed.

```
ELAPCB    [WORKDBD=customer-dbd-name]

--- expands into ---

PCB      TYPE=DB,DBDNAME=customer-dbd-name,PROCOPT=AP,KEYLEN=19
SENSEGE  NAME=ELAWCNTL,PARENT=0
SENSEGE  NAME=WORKLV01,PARENT=ELAWCNTL
SENSEGE  NAME=WORKLV02,PARENT=WORKLV01
:
:
SENSEGE  NAME=WORKLV14,PARENT=WORKLV13
SENSEGE  NAME=MSGLV01,PARENT=ELAWCNTL
SENSEGE  NAME=MSGLV02,PARENT=MSGLV01
:
:
SENSEGE  NAME=MSGLV14,PARENT=MSGLV13
```

Figure 7. Generating the DL/I Work Database PCB

Migrating from CSP/370RS V1R1

The ELAPCB macro for VisualAge Generator Server supports only the work database. PSBs for programs generated using CSP/370RS V1R1 might also have used a message database. If you need to generate the PSB for these programs again, you must use the ELAPCB macro from CSP/370RS V1R1.

Processing Modes

IMS requires segmented or single-segment mode. Refer to the *VisualAge Generator Design Guide* for additional information on segmented and single-segment modes.

Whether a program runs as IMS conversational or nonconversational is determined at generation time by using the /SPA generation option. Refer to the *VisualAge Generator Generation Guide* for information on how to specify the /SPA generation option.

The work database is used for both conversational and nonconversational processing to save information during a CONVERSE process option. In conversational mode, the scratch-pad area (SPA) is used to set the transaction identifier and to save information during a program-to-program message switch. Refer to the *VisualAge Generator Design Guide* for information on how the SPA is used for program-to-program message switching.

Printing Considerations for IMS

From VisualAge Generator Server, printing is initiated when a program processes a DISPLAY process option for a VisualAge Generator-defined printer map. Refer to the online helps for map definition within VisualAge Generator Developer for information on defining maps for printers.

Printing is accomplished using MFS control blocks produced when the map group is generated. The default print destination in IMS is a message queue named EZEPRINT. The printer destination can be changed at generation time. You can also change the print destination at run time by using the EZEDESTP special function word. Refer to the *VisualAge Generator Design Guide* or to the VisualAge Generator help facility for additional information.

Recovery and Database Integrity Considerations

VisualAge Generator programs can make use of all the IMS recovery and data integrity features.

If your program makes changes to a recoverable resource, the changes are not committed until the end of a logical unit of work (LUW). If your program abnormally ends before the end of an LUW, all changes that were made since the beginning of the LUW are backed out.

An LUW for an IMS transaction ends whenever a commit point or a rollback occurs. A commit point occurs in IMS when one of the following occurs:

- The top-level program in a run unit ends successfully.
For IMS BMPs, a run unit consists of all VisualAge Generator programs and non-VisualAge Generator programs that transfer control among themselves using an XFER, DXFR, or CALL statement. For non-VisualAge Generator programs, this also includes transfers using an OS XCTL macro or CALL statement.
For IMS/VS, a run unit is equivalent to a single transaction and consists of all VisualAge Generator programs and non-VisualAge Generator programs that transfer control among themselves using a DXFR or CALL statement. For non-VisualAge Generator programs, this also includes transfers using a CALL statement.
- A program uses a CONVERSE process option.
The best time for a commit point to occur is after terminal output and before the next terminal input. A commit point at terminal I/O synchronizes updates to the database and confirmation messages to the program user.
- A batch-oriented IMS BMP program (one that does not scan a serial file associated with the I/O PCB) calls the EZECOMIT service routine.
- A batch-oriented IMS BMP program uses an XFER statement and the /SYNCXFER generation option was specified for the transferred-from program.
- A program does a successful get unique to the I/O PCB.

A rollback occurs when one the following occurs:

- A program calls the EZEROLLB service routine.
- A program ends because of an error condition.

When a rollback occurs, all database changes that were made since the start of the LUW are backed out.

Considerations that Affect Performance

This section describes factors that affect system performance and suggestions on how to improve performance.

Residency Considerations and the IMS Preload Function

The performance of a program is affected by the number of times a disk is accessed while running the program. Programs require access to disks for the following reasons:

- Locating and loading VisualAge Generator Server load modules
- Retrieving and storing user data
- Locating and loading application, map group format modules, MFS print services programs, and table load modules

VisualAge Generator Server loads objects as they are needed. For example, VisualAge Generator Server loads an application, MFS print services program, map group format module, or table when another program calls or references it. The overhead of locating and loading modules can be reduced by using the IMS preload function. Preloading an object reduces the amount of I/O required for multiple loads. However, preloading generated programs requires more virtual storage for your system because preloaded modules remain in storage until the message region is shut down.

It is usually not possible for everything to be preloaded. Therefore, you should establish priorities for deciding which objects you should preload. These preloading priorities reflect a trade-off between your program usage and your storage constraints. Because of individual considerations such as storage constraints, environment, and types of programs, your priorities might dictate that some components (such as MFS print services programs) for a program be preloaded, while other components (such as tables) should not be preloaded. Make the decision on what modules to preload on an individual basis, according to how the program uses them.

When deciding what to preload, consider the following:

- Storage constraints
- Frequency of program use
- Long-running programs as compared to programs that are started more frequently

Generally, objects that are loaded more than once are prime candidates for preloading. Examples of this are a table that is used by more than one program and a program that is called more than one time. The following are some general rules for preloading:

- When deciding what to preload, consider the following objects:
 - Called programs
 - MFS print services programs
 - Map group format modules
 - Tables

- Main programs
- Programs that are started or referenced frequently should be preloaded. In addition to programs that are loaded by IMS when a transaction is scheduled, this includes programs that are started by the VisualAge Generator CALL and DXFR statements.
- Programs that are not frequently initiated should not be preloaded.

See “Preloading Generated Programs” on page 94 for additional information.

Preloading VisualAge Generator Server Modules

For best performance, use the preload option for the following VisualAge Generator Server modules:

- ELARPRTR, the VisualAge Generator Server module that handles address mode switching
- ELARPRTM, the VisualAge Generator Server load module
- ELARPIOP, the installation options module
- ELARlccc (where ccc is the language code), the language-dependent options module
- ELACNccc (where ccc is the language code), the conversion table
- ELANCccc (where ccc is the language code), the module for VisualAge Generator Server constants and the fold table
- ELARSCNT, the configuration table
- ELA2SSQW, the module that supports the DB2 work database
- ELARSDCB, which is used for accessing VisualAge Generator Server sequential files
- ELA2SSQL, its alias ELA2SSQY, and ELA2SSQX
 ELA2SSQL, its alias ELA2SSQY, and ELA2SSQX are used to gain access to the DB2 work database, and they support commit and rollback processing for DB2 program databases. Preload these modules only if you are using programs that were generated and bound using CSP/370RS V1R1.

The modules ELARSDCB and ELANCccc are loaded below the 16MB line. ELARSDCB is used only in reporting errors detected by VisualAge Generator Server. Both can be omitted from the preload list if storage space below the 16MB line is limited.

Note: You should also monitor the usage of the LE runtime modules. Because many are used by the generated COBOL programs, these modules might also be candidates for preloading.

Refer to the IMS documentation for your system for information on the preload option. An alternative to preloading is to place modules in the link pack area.

Loading VisualAge Generator Server Modules into the Link Pack Area

Placing modules in the link pack area causes all regions to share a single copy of the modules and saves storage space. Refer to the *Program Directory for VisualAge Generator Server for MVS* (GI10-0812) for information about what modules can be put into the link pack area.

Only one version of CSP/370RS V2R1, or CSP/370RS V1R1, or IBM VisualAge Generator Server for MVS, VSE, and VM modules can be placed in the link pack area. If multiple releases are installed concurrently on the same system, override the link pack area by defining the correct library in the STEPLIB or JOBLIB DD statements for the region.

Preloading Generated Programs

You can reduce the overhead of searching the STEPLIB, JOBLIB, link pack area, and link list by preloading generated programs (application programs, online print services programs, map group format modules, and table modules) that are frequently used. However, in this case, virtual storage is still occupied by the modules when they are not in use.

To improve response time, you might also preload any module associated with any transaction that might require better performance, even though the module itself is not frequently used.

To preload generated programs, do the following:

1. Put the module in a LNKLST library.
2. Include the module name in a preload member (DFSMPLEX, where xx is a two-character ID that you select) in the IMS procedure library.
3. Indicate in the JCL for the IMS message region that the preload member is to be included.

Database Performance

Database performance can be improved under IMS/ESA by defining HIPERSPACE* buffer usage for IMS in the DFSVSMxx member. This is the same as defining many buffers for the files, but has the advantage that the HIPERSPACE buffers all come from 31-bit storage, not from within the IMS/ESA region. The tuning of database buffer pools is recommended. Refer to the IMS manuals for details on the tuning of database buffer pools.

If you have IMS/ESA installed and use a DL/I work database, make the work database nonrecoverable to reduce the amount of logging that occurs. Making the work database nonrecoverable might help improve performance.

Limiting MFS Control Blocks

Limiting the size and number of message format service (MFS) control blocks might help improve performance. MFS is used for mapping support in the IMS environment. MFS control blocks are generated using MFS utility control statements.

You can reduce the size and number of MFS control blocks that are generated by doing the following:

- In device selection of map definition, select only those VisualAge Generator devices that are used for the application system. For additional information about valid device types that can be specified, refer to the Devices list on the Map Definition Profile - Device Selection window within VisualAge Generator Developer.
- Include in the /MFSDEV generation option only device types that your installation or application system uses. For additional information about specifying the /MFSDEV generation option, see 107.

Monitoring and Tuning the IMS System

You can track potential performance problems before they occur by checking processing statistics on a regular basis. The following are some of the statistics to monitor:

- Use the IMS DC monitor facilities to check transaction utilization. Consider preloading programs or groups of programs which are frequently used.
- Use the IMS database monitor facilities to check how effectively the databases are performing and using space.

You can also use the following tools to monitor IMS performance:

- The IMS Performance Analysis and Reporting System (IMSPARS, Program No. 5798-CQP). This tool presents information on transaction transit times, IMS resource usage, and IMS resource availability, as well as detailed reports tracing individual transaction and database change activity. These reports are based on the contents of the IMS log data set.
- The Resource Measurement Facility* (RMF*) II . This tool collects information about processor, channel, and I/O device utilization.
- The DB Tools product (Program No. 5685-093). This tool provides information to help improve database efficiency and space utilization.

Refer to the system administration manuals and the database administration guide for your release of IMS for detailed information about monitoring the IMS online system and DL/I databases.

Considerations for Using DB2 in IMS

This section discusses considerations for recovery, database integrity, and security issues for DB2 programs.

For information on designing and generating DB2 programs for the IMS environment, refer to the *VisualAge Generator Design Guide* and the *VisualAge Generator Generation Guide*.

For information on preparing DB2 programs for running in the IMS environment, see “Chapter 18. Preparing and Running Generated Programs in IMS/VS and IMS BMP” on page 165.

Recovery and Database Integrity Considerations

VisualAge Generator programs can use all the recovery and data integrity features that are provided by DB2 in the IMS environment.

Relational databases are recoverable resources. If your program makes changes to a relational database, the changes are not committed to the database until the end of a logical unit of work (LUW). If your program ends abnormally before the end of an LUW, all changes that were made since the beginning of the LUW are backed out. See “Recovery and Database Integrity Considerations” on page 90 for additional information on when an LUW ends.

Checking Authorization

The database manager checks whether the program users have authority to gain access to tables or to run programs. The type of checking done varies depending on your system and the processing mode.

When using DB2 in generated COBOL programs, the program users must be authorized to run the corresponding DB2 program plan. For transaction-oriented regions, the authorization ID depends on the type of IMS security being used:

- If sign-on security is used, IMS provides the sign-on name as the authorization ID.
- If sign-on security is not used, IMS provides the name of the originating terminal as the authorization ID.

The plan used with a transaction has the same name as the program associated with the transaction.

For batch-oriented regions, the authorization ID is the contents of the ASXBUSER field, if valid, or the PSB name. The plan name is specified as one of the batch program parameters.

For more information on IMS security mechanisms, refer to the appropriate IMS manual.

Considerations for Using DL/I in IMS

This section discusses considerations for DL/I programs in the IMS environment.

See “Defining the Program Specification Block (PSB)” on page 88 for information on defining a PSB for DL/I programs.

For information on designing and generating DL/I programs for the IMS environment, refer to the *VisualAge Generator Design Guide* and the *VisualAge Generator Generation Guide*.

For information on preparing DL/I programs for running in the IMS environment, see Chapter 18. Preparing and Running Generated Programs in IMS/VS and IMS BMP.

Recovery and Database Integrity Considerations

VisualAge Generator programs can make use of all the recovery and data integrity features that are provided for DL/I databases in the IMS environment.

DL/I databases are recoverable resources. If your program makes changes to a DL/I database, the changes are not committed to the database until the end of a logical unit of work (LUW). If your program ends abnormally before the end of an LUW, all changes that were made since the beginning of the LUW are backed out. See “Recovery and Database Integrity Considerations” on page 90 for additional information on when an LUW ends.

Maintaining the Work Database in IMS

You should monitor and tune the DL/I and DB2 work databases just as you would any other DL/I database or DB2 table. You can use the normal database administration utilities to monitor these databases and to determine when they need to be reorganized to improve performance.

The activities involved in maintaining the work database are the following:

- Deleting old records from the work database
- Expanding the work database
- Supporting multiple DL/I or DB2 work databases

Deleting Old Records from the Work Database

The terminal ID is the key for the records in the work database. Each record contains a time stamp that indicates the last time the record was updated.

Deleting old records from the database reduces the amount of disk space required in the work database. You probably want to delete records in the following situations:

- Some users might run a generated program only infrequently, less than once a day, for example. In this case, you might want to delete old records on a daily or weekly basis.
- Sometimes terminal names are changed or users are moved to terminals with different names. In this case, new records are created for the new terminals, but the old records are not automatically deleted.

The utilities that delete records from the DL/I and DB2 work databases validate the date and time to ensure that your request does not result in deletion of records that are less than 24 hours old.

DL/I Work Database

Figure 8 on page 99 shows the JCL used to remove old records from a DL/I work database. The JCL is supplied as member ELAWKJCD in the ELA.V1R2M0.ELAJCL file. Specify the records you want to delete by entering the date (in Julian format) and time prior to which all records are to be deleted.

```

//*****
//** ELAWKJCD - JOBSTREAM TO CLEAN UP THE DLI WORK DATABASE
//**          FOR VISUALAGE GENERATOR SERVER.
//**
//** LICENSED MATERIALS - PROPERTY OF IBM
//** 5648-B02 (C) COPYRIGHT IBM CORP. 1994, 1998
//** SEE COPYRIGHT INSTRUCTIONS
//**
//** STATUS = VERSION 1, RELEASE 2, LEVEL 0
//**
//** TO TAILOR THIS JOBSTREAM:
//**     1. COPY A JOBCARD.
//**     2. REPLACE DATE AND TIME STAMP VALUE WITH DESIRED
//**        VALUE. ALL RECORDS WITH LESS THAN THAT DATE AND
//**        TIME WILL BE DELETED.
//**
//** RETURN CODES
//**     0 - SUCCESSFUL COMPLETION
//**    12 - FATAL ERROR. INVALID INPUT
//**    16 - FATAL ERROR. PROCESSING TERMINATED
//**
//*****
//*
//DLIWORK      EXEC IMSBATCH,MBR=ELAWKPC1,
//              PSB=ELAWKPB1,RGN=4096K
//G.STEPLIB    DD
//              DD
//              DD DSN=CEE.SCEERUN,DISP=SHR
//              DD DSN=ELA.V1R2M0.SELALMD,DISP=SHR
//G.ELAPRINT   DD SYSOUT=*
//G.SYSOUT     DD SYSOUT=*
//G.SYSIN      DD *
YYDDHHMMSS

```

Figure 8. JCL to Remove Old Records from DLI Work Database

DB2 Work Database

Figure 9 on page 100 shows the JCL used to remove old records from a DB2 work database. The JCL is supplied as member ELAWKJC2 in the ELA.V1R2M0.ELAJCL file. Specify the records you want to delete by entering the date (in Julian format) and time prior to which all records are to be deleted.

```

//*****
/** ELAWKJC2 - JOBSTREAM TO CLEAN UP THE DB2 WORK DATABASE
/**          FOR VISUALAGE GENERATOR SERVER.
/**
/** LICENSED MATERIALS - PROPERTY OF IBM
/** 5648-B02 (C) COPYRIGHT IBM CORP. 1994, 1998
/** SEE COPYRIGHT INSTRUCTIONS
/**
/* STATUS = VERSION 1, RELEASE 2, LEVEL 0
/**
/** TO TAILOR THIS JOBSTREAM:
/**      1. COPY A JOBCARD.
/**      2. REPLACE DATE AND TIME STAMP WITH THE DESIRED DATA.
/**          ALL ROWS WITH A DATE AND TIME LESS THAN THE
/**          SPECIFIED DATE/TIME WILL BE DELETED.
/**
/** RETURN CODES
/**      0 - SUCCESSFUL COMPLETION
/**      12 - FATAL ERROR. INVALID INPUT
/**      16 - FATAL ERROR. PROCESSING TERMINATED
/**
//*****
/*
//DB2WORK EXEC PGM=ELAWKPC2,REGION=4096K
//STEPLIB DD DSN=CEE.SCEERUN,DISP=SHR
//          DD DSN=ELA.V1R2M0.SELALMD,DISP=SHR
//SYSOUT DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//ELAPRINT DD SYSOUT=*
//ELASNAP DD SYSOUT=*
//EZESPUFI DD DSN=&&TMP1,DISP=(NEW,PASS),
//          UNIT=SYSDA,SPACE=(TRK,(1,0)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=80)
//SYSIN DD *
YYDDHHMMSS
/*
//DB2SPUF EXEC PGM=IKJEFT01,REGION=4096K,COND=(0,NE)
//STEPLIB DD DSN=DSN.RUNLIB.LOAD,DISP=SHR
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD DSN=&&TMP1,UNIT=SYSDA,DISP=(OLD,DELETE)

/*
//SYSTSIN DD *
DSN SYSTEM(DSN)
RUN PROGRAM(DSNTIAD) PLAN(DSNTIA??)
END
/*

```

Figure 9. JCL to Remove Old Records from DB2 Work Database

Expanding the Work Database

At times, you need to expand the work database. For example, you need to expand the database when you expand the usage of an existing program system to a larger user set comprising a much larger number of terminals that gain access to VisualAge Generator programs.

DL/I Work Database

To expand the DL/I work database, perform the following steps:

1. Stop the DL/I database.
2. Unload the database using the old database description (DBD).
3. Change the DBD information and perform a DBD generation.
4. If you are having application control blocks (ACBs) prebuilt rather than built dynamically, build the ACBs again.
5. Delete the space allocated for the old database and allocate space for the new definition.
6. Load the database using the new DBD.
7. Make an image copy of the new database for back-up purposes as soon as it is loaded.

Refer to the database administrator's guide and the IMS utilities manual for additional information.

DB2 Work Database

You might need to expand the table spaces containing the DB2 work database because of degraded performance from too many secondary extents, or because the application users receive a DB2 message DSNP007I indicating that no more space is available.

Ideally, when the size of a DB2 table space is increased, the primary extent should be made large enough to accommodate all the data in the work database. In any case, try to minimize the number of secondary extents required to store rows in the database.

The method you use to expand the table space depends on the version of DB2 that is installed and whether the table space is user-managed.

The procedure supplied with VisualAge Generator Server that installs the work database also installs the table space as user-managed table space (no associated DB2 storage group).

Before attempting to change the size of the table space data set, you need to estimate the space requirements for the table space. One factor in your

estimate is the amount of space currently used. If the space is currently DB2-managed (resulting from an earlier change in space allocation), you can get this information by first running the DB2 STOSPACE utility against the table space storage group, and then running the following query:

```
SELECT SPACE
FROM SYSIBM.SYSTABLEPART
WHERE TSNAME='tsname' and DBNAME='dbname';
```

The result (SPACE) gives the number of kilobytes of storage currently allocated to the table space.

If the space for the table space is user-managed, you can use the MVS/TSO LISTCAT command to obtain the space information. You need to know the data set name of the VSAM file used for table space. The data set name for the VSAM file has the following format:

```
catname.DSNDBC.dbname.tsname.I0001.Annn
```

where:

catname	Specifies the VSAM catalog name or alias This is the same name or alias as in the USING VCAT clause of the CREATE TABLESPACE statement.
dbname	Specifies the DB2 database name This is the same as the database name in the CREATE TABLESPACE statement.
tsname	Specifies the table space name This is the same as the table space name in the CREATE TABLESPACE statement.
nnn	Specifies the data set number For partitioned table spaces, the number is 001 for the first partition, 002 for the second, and so forth, up to the maximum of 64 partitions. For a simple or segmented table space, the number is 001 for the first data set. If the simple or segmented table space exceeds 2 gigabytes, the second data set is 002, and so forth.

To expand table space do the following:

1. Stop the DB2 database by using the command -STOP DB (dbname).
2. Make an image copy of the table space. You can use the image copy to restore the data set if the procedure is not successful.
3. Create a storage group for the table space. Do this only if the table space currently is user-managed and a storage group is not already available.

4. Change the table space definition as follows:

- If the table space data sets are user-managed, use a DB2 statement as follows:

```
ALTER TABLESPACE dbname.tsname
  USING STOGROUP stogrp
  PRIQTY pppp SECQTY ssss
```

where:

dbname.tsname	Specifies the name of the space
stogrp	Specifies the name of the storage group
pppp	Specifies new primary allocation size (in kilobytes) for the expanded table space
ssss	Specifies new secondary allocation size (in kilobytes) for the expanded table space

Note: This statement changes the table space from user-managed to DB2-managed.

- If the table space data sets are already DB2-managed, use a DB2 statement as follows:

```
ALTER TABLESPACE dbname.tsname
  PRIQTY pppp SECQTY ssss
```

where:

dbname.tsname	Specifies the name of the space
pppp	Specifies new primary allocation size (in kilobytes) for the expanded table space
ssss	Specifies new secondary allocation size (in kilobytes) for the expanded table space

5. Move the table space data. Simply changing the table space definition does not put the new size into effect. You need to move the table space to the newly allocated space. You can, for example, reorganize the table space using the DB2 REORG utility.
6. Start the DB2 database. Enter the command `-START DB (dbname)`.

Supporting Multiple Work Databases

You can use separate work databases for different application systems. For example, you might want to use separate databases for payroll and shipping to improve performance or to increase data availability. The work database is used to pass information during certain types of program-to-program message

switches between applications. When this occurs, both the transferring application and the transferred-to application must use the same physical work database.

DL/I Work Databases

To create an additional DL/I work database called ELAWORK2, do the following:

1. Copy the ELAWORK DBD in the ELA.V1R2M0.ELASAMP file, and name it ELAWORK2.
2. Change the NAME parameter on the DBD statement to ELAWORK2. Also change the DD1 parameter on the DATASET statement to ELAWORK2. Make any other changes to the block size, number of blocks, and randomizing routine based on the application system requirements.
3. Make copies of the ELAWKLD and ELAWKPB1 program specification blocks (PSBs) in the ELA.V1R2M0.ELASAMP file and give them new member names. Change the NAME parameter on the program control block (PCB) statement from ELAWORK to ELAWORK2.
4. Modify job ELACJWKD in the ELA.V1R2M0.ELAJCL file to refer to the new database. This job does the DBD, PSB, and ACB generations needed for the work database, allocates the database, and then initializes it. You need to change the DD and data set names for the work database, and name the new DBD and PSB.
5. Add the new database to the JCL for your IMS control region, and to your IMS stage-1 system definition.
6. When you create IMS PSBs for applications that need to use this new database, use the ELAPCB macro to create the PCB definition for the work database. Enter the following command:

```
ELAPCB WORKDBD=ELAWORK2
```

DB2 Work Databases

To create an additional DB2 work database, do the following:

1. Create an ELAWORK table using the ELACJWK2 job in the ELA.V1R2M0.ELAJCL file. Perform the following steps before running the job:
 - a. Add an authorization ID to the CREATE TABLE command in ELAWORK2 in the ELA.V1R2M0.ELASAMP file, for example:

```
CREATE PAYROLL.ELAWORK
```
 - b. Change the table space name and index in ELAWORK2.
 - c. Change the DELETE and DEFINE CLUSTER statements to use the table space name and index you specified in ELAWORK2.

- d. Comment out the WRKDROP step to avoid dropping the existing work database.
2. Each developer or system administrator using the payroll ELAWORK table needs to create a SYNONYM for the table. The following example shows how to use the CREATE SYNONYM command to create a synonym:

```
CREATE SYNONYM ELAWORK FOR PAYROLL.ELAWORK
```

The BIND command generated from the default BIND templates EFK0MBDB, EFK0MBDD, and EFK0MBDC bind DBRMs for VisualAge Generator Server modules to the application being generated. The CREATE SYNONYM command ensures that developers referencing the ELAWORK table use the payroll version of the table.

Considerations for Message Format Services in IMS

VisualAge Generator generates message format services (MFS) source statements used for conversing and displaying maps in IMS environments. The generated MFS source includes DEV statements, which identify the device types on which maps can be displayed and the characteristics of those devices. The device types and characteristics must be compatible with the device types and characteristics defined in the TERMINAL and TYPE macros in your IMS system definition.

The information on the generated DEV statements is controlled by the /MFSDEV, /MFSEATTR, /NOMFSEATTR, /MFSEATTRNCD, /MFSIGNORE, and /NOMFSIGNORE generation options. Review the default specifications for the TERMINAL and TYPE and change them to be compatible with your system definition. You can delete the /MFSDEV entries for any terminal types not supported at your installation and change the DEV statement parameters for any types for which the default parameters are not correct.

VisualAge Generator provides sample generation option files that affect the generation of MFS source:

- EFKOPIMS.OPT for the IMS/VS target environment
- EFKOPBMP.OPT for the IMS BMP target environment

You can edit the options files using a standard OS/2* editor. The following figure shows how the entries in the /MFSDEV option can be split into multiple lines.

```

/MFSDEV=('ANY-D', 'TYPE=3270-A8,FEAT=(IGNORE)',EATTR),1
('ANY-2D', 'TYPE=3270-A2,FEAT=(IGNORE)',EATTR)2

```

Notes:

1. The entries can be separated at a comma following each device pair.
2. The next device pair starts with a left parenthesis on the new line.

Figure 10. How to Split MFSDEV Entries into Multiple Lines

The following generation options affect the generated MFS source:

/NOMFSEATTR	<p>Specifies that extended attributes are not used</p> <p>Specify /NOMFSEATTR if most of your devices do not support color or extended highlighting. /NOMFSEATTR specifies that the EATTR parameter is omitted from the MFS DFLD statements unless overridden by the /MFSDEV option for a specific device. The default is /MFSEATTR.</p>
/MFSEATTR	<p>Specifies the default handling for extended attributes on the MFS DFLD statement</p> <p>Specify /MFSEATTR if all of your devices support extended attributes (for example, devices that support color or extended highlighting), and you want the CD (color default) extended attribute value generated when a map field is defined as MONO (monochromatic). /MFSEATTR specifies that the EATTR parameter is included for the DFLD statements unless overridden by the /MFSDEV option for a specific device.</p>
/MFSEATTRNCD	<p>Specifies that you do not want the CD (color default) extended attribute value generated for maps defined as MONO</p> <p>Specify /MFSEATTRNCD if your devices support extended attributes, but you do not want the CD extended attribute value generated for maps defined as MONO. /MFSEATTRNCD specifies that EATTR is generated, and CD is not used for fields specified as MONO. The default is /MFSEATTR.</p>

/MFSIGNORE

Specifies information used in the MFS MSG statement

Specify /MFSIGNORE if you want SOR=(...,IGNORE) to be included on the MSG statement for the message input descriptor (MID) and message output descriptor (MOD). Specify /MFSIGNORE only if the /MFSDEV generation option specifies FEAT=IGNORE for all of the devices used by the map group you are generating. The default is /NOMFSIGNORE.

/NOMFSIGNORE

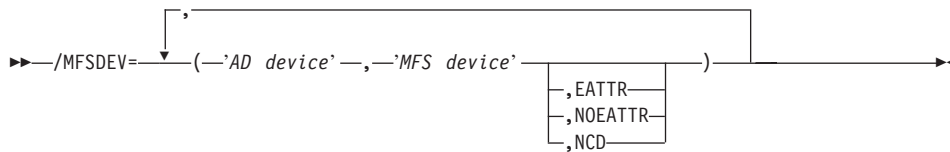
Specifies that you do not want to include IGNORE for the SOR parameter on the message statement for the MID and MOD in the generated message format service (MFS)

/MFSDEV

Specifies information used in the MFS DEV statement

Note: The /MFSEATTR, /MFSEATTRNCD, and /NOMFSEATTR generation options specify how the DFLD statements for a specific device are to be generated if the EATTR, NCD, or NOEATTR parameter is not included for the device in the /MFSDEV generation option. If EATTR, NCD, or NOEATTR is specified for a device in the /MFSDEV option, the /MFSEATTR, /MFSEATTRNCD, and /NOMFSEATTR options have no effect for that device.

The /MFSDEV option associates the VisualAge Generator device types that application developers specify during map definition with the actual DEV statement options that are included in the generated MFS source. The following shows the format for the /MFSDEV option. See "Appendix A" for more information about the conventions for syntax diagrams.



where:

AD device

Is any valid device type that can be specified during map definition.

For additional information on the valid device types that can be specified, refer to the

Devices list on the Map Definition Profile - Device Selection window in VisualAge Generator Developer.

MFS device

Specifies one or more of the parameters used on a DEV statement for MFS.

Enclose the string of parameters in single quotes. The information in the MFS device variable is based on the information supplied for the TERMINAL and TYPE macros in your IMS system definition.

EATTR, NOEATTR, or NCD

Indicates whether the device supports extended attributes.

They are optional and allow you to override the default values you specified for the /MFSEATTR, /NOMFSEATTR, or /MFSEATTRNCD generation options. Specify EATTR, NOEATTR, or NCD for a specific device only if the value differs from the default value specified by your choice among the /MFSEATTR, /NOMFSEATTR, and /MFSEATTRNCD generation options. If you specify EATTR, the EATTR parameter is included for the DFLD statements, and CD is used for fields specified as MONO. If you specify NOEATTR, no EATTR parameter is generated. If you specify NCD, the EATTR parameter is generated, but CD is not used for fields specified as MONO.

Note: The number of pairs of device types and MFS device parameters that you specify with the /MFSDEV generation option should be limited to those devices actually used in your installation.

The default value for the /MFSDEV generation option is as follows:

```
/MFSDEV=('ANY-D', 'TYPE=3270-A8, FEAT=(IGNORE)', EATTR),
        ('ANY-2D', 'TYPE=3270-A2, FEAT=(IGNORE)', EATTR),
        ('ANY-2D', 'TYPE=(3270,2), FEAT=(IGNORE)', EATTR),
        ('ANY-3D', 'TYPE=3270-A3, FEAT=(IGNORE)', EATTR),
        ('ANY-4D', 'TYPE=3270-A4, FEAT=(IGNORE)', EATTR),
        ('ANY-5D', 'TYPE=3270-A7, FEAT=(IGNORE)', EATTR),
        ('PRINT-B', 'TYPE=3270P, WIDTH=133, PAGE=(255,DEFN), FEAT=2', EATTR),
        ('PRINT-B', 'TYPE=SCS1, WIDTH=132, PAGE=(255,DEFN)', EATTR),
        ('PRINTER', 'TYPE=3270P, WIDTH=133, PAGE=(255,DEFN), FEAT=2', EATTR),
        ('PRINTER', 'TYPE=SCS1, WIDTH=132, PAGE=(255,DEFN)', EATTR),
        ('3278-2B', 'TYPE=3270-A2, FEAT=(IGNORE)', EATTR),
        ('3278-2B', 'TYPE=(3270,2), FEAT=(IGNORE)', EATTR),
```

```

('3278-3B', 'TYPE=3270-A3, FEAT=(IGNORE)', EATTR),
('3278-4B', 'TYPE=3270-A4, FEAT=(IGNORE)', EATTR),
('3278-5B', 'TYPE=3270-A7, FEAT=(IGNORE)', EATTR),
('3278-2', 'TYPE=3270-A2, FEAT=(IGNORE)', EATTR),
('3278-2', 'TYPE=(3270,2), FEAT=(IGNORE)', EATTR),
('3278-3', 'TYPE=3270-A3, FEAT=(IGNORE)', EATTR),
('3278-4', 'TYPE=3270-A4, FEAT=(IGNORE)', EATTR),
('3278-5', 'TYPE=3270-A7, FEAT=(IGNORE)', EATTR),
('3767', 'TYPE=3270P, WIDTH=133, PAGE=(255,DEFN), FEAT=2', EATTR),
('3767', 'TYPE=SCS1, WIDTH=132, PAGE=(255,DEFN)', EATTR),
('5550D', 'TYPE=3270-A2, FEAT=(IGNORE)', EATTR),
('5550P', 'TYPE=SCS1, WIDTH=158, PAGE=(255,DEFN)', EATTR)

```

The defaults for MFS device variables include FEAT=IGNORE for display devices. If the IMS system definition also includes FEAT=IGNORE for the TERMINAL or TYPE macros, you might be able to use the defaults as shown. However, you should still review the defaults to eliminate device types that your installation does not support. Removing unused device entries makes the MFS control blocks smaller and might help improve performance.

Table 14 shows the relationship between the /MFSIGNORE, /NOMFSIGNORE, and /MFSDEV generation options and the FEAT parameter for the TERMINAL and TYPE macros in the IMS system definition.

Table 14. Relationship between /MFSIGNORE, /NOMFSIGNORE, /MFSDEV, and IMS System Definition

Generation Option	MFS MSG Statement for MID / MOD	/MFSDEV FEAT Parameter	IMS System Definition FEAT Parameter
/MFSIGNORE	SOR=(xxxxxx,IGNORE)	FEAT=IGNORE ¹	FEAT=IGNORE or FEAT=n
/MFSIGNORE ²		FEAT=n ²	
/NOMFSIGNORE	SOR=xxxxxx	FEAT=IGNORE ³	FEAT=IGNORE
/NOMFSIGNORE	SOR=xxxxxx	FEAT=n ³	FEAT=n

Note:

1. The value for FEAT in the /MFSDEV option does not need to match the value for FEAT in the IMS TERMINAL or TYPE macro.
2. This combination of the /MFSIGNORE and /MFSDEV options is not valid. Generation ignores any device that uses this combination because the combination is not supported by MFS.
3. The value for FEAT in the /MFSDEV option must exactly match the value specified for FEAT in the IMS TERMINAL or TYPE macro.

Table 15 on page 110 shows parameters from the TERMINAL and TYPE macros in your IMS system definition that you can code for the MFS device

variable. Other MFS parameters for the DEV statement should not be coded for the MFS device variable.

Table 15. Parameters for MFS Device Specification

Description	/MFSDEV TYPE Parameter	Optional Parameters
3270 Display	(3270,1), (3270,2), 3270-An ¹	FEAT
3270 Printer	3270P	FEAT, WIDTH ² PAGE ³
SCS1 Printer	SCS1	FEAT, WIDTH ⁴ PAGE ³

Note:

1. The n in 3270-An is any number from 1 through 15.
2. If WIDTH is coded FEAT must be coded. WIDTH must be a value 1 greater than the width for the AD device because the last column is used by MFS for carriage control. To have compatibility for the AD device values 3767, PRINTER, or PRINT-B, use FEAT=n (where n is a value from 1 through 10 and matches your IMS system definition), WIDTH=133, PAGE=(255,DEFN).
3. If PAGE is coded and the second parameter is given, it must be DEFN. DEFN is the default.
4. To have compatibility for the AD device values 3767, PRINTER, or PRINT-B, use WIDTH=132, PAGE=(255,DEFN). To have compatibility for an AD device value of 5550P, use WIDTH=158, PAGE=(255,DEFN).

Refer to the IMS system definition reference manual for your release of IMS for additional information on the parameters for the TERMINAL and TYPE macros. Also refer to the stage-1 system definition macros for your IMS system to determine the parameters actually used for your installation. Refer to the MFS manuals for your release of IMS for additional information about the DEV statement.

You might also want to look at your non-VisualAge Generator MFS source to see the parameters that you currently specify on the DEV statement.

Once you have determined the correct values for /MFSDEV, /MFSEATTR, and /MFSIGNORE options, code the default generation options in all the default generation options files that you use when generating for the IMS or IMS BMP target environments.

Part 3. Preparing and Running Generated Applications on MVS Systems

Chapter 12. Output of Program Generation on MVS Systems	113
List of Program Preparation Steps after Program Generation.	113
Outputs of Generation	114
Objects Generated for Programs.	117
Application COBOL Program	117
Sample Runtime JCL	118
Sample Runtime CLIST	118
BIND Commands	119
Objects Generated for Tables	119
Table COBOL Program	119
Objects Generated for Map Groups.	119
Online Print Services Program	119
Message Format Service (MFS) Print Services Program.	119
Batch Print Services Program.	119
Map Group Format Module	120
MFS Control Blocks	120
COBOL Copybook for MFS	
MID/MOD Layout	121
Objects Generated for All Part Types	121
Listing File	121
Parts Control File	121
Preparation Command File	121
Preparation JCL for MVS	
Environments	121
MVS CICS Program and Transaction Definitions	122
Chapter 13. Preparing Generated Programs on MVS Systems.	123
Allocating Preparation Data Sets	123
Running the Preparation JCL.	126
Analyzing the Preparation Listing	126
Analyzing the DB2 Precompile Step	127
Analyzing the CICS Translator Step	127
Analyzing the Compile Step	127
Analyzing the Link-Edit Step.	127
Link-Edit Errors Related to Missing Modules	128
Other Link-Edit Errors	128
Analyzing the BIND Step	128
Preparing to Run a DB2 Program	129
Granting Authority to Program Users	129
Defining Synonyms for DB2 Tables.	129
Using LE Runtime Options for MVS/TSO and MVS Batch	130
Chapter 14. Creating or Modifying Runtime JCL or CLISTs on MVS Systems	135
Tailoring JCL or CLIST Templates before Generation	135
Modifying Runtime JCL	137
Modifying Runtime CLISTs	139
Chapter 15. Preparing and Running a Generated Program in MVS CICS.	143
Modifying CICS Resource Tables	143
Program Entries (PPT)	144
Transaction Entries (PCT)	145
Destination Control Table Entries (DCT)	147
File Control Table Entries (FCT)	147
Resource Control Table Entry (RCT)	148
Using Remote Programs, Transactions, or Files	148
Using MVS JES Spool Files	148
Modifying CICS Startup JCL	148
Making New Modules Available in the MVS CICS Environment	148
Making Programs Resident	149
Running Programs under MVS CICS	149
Starting the Transaction in MVS CICS	150
Controlling Diagnostic Information in the CICS Environment	150
Printing Diagnostic Messages in the CICS Environment	150
Chapter 16. Preparing and Running a Generated Program in MVS/TSO	151
Making VisualAge Generator Server, COBOL, and the Generated Modules Available	151
Running Main Programs under MVS/TSO	152
Starting the CLIST	152
Examples of Runtime CLISTs for MVS/TSO	153

Running a Main Program with No Database Access	153	Creating MFS Control Blocks.	168
Running a Main Program with DB2 Access	154	Making New Modules Available in the IMS Environment	169
Running a Main Program with DL/I Access	156	Preloading Program, Print Services, and Table Modules	169
Chapter 17. Preparing and Running Generated Programs in MVS Batch	159	Running Programs under IMS	170
Running Main Programs under MVS Batch	159	Starting a Main Program Directly	170
Examples of Runtime JCL for MVS Batch Programs	160	Starting a Main Transaction Program	
Running a Main Batch Program with No Database Access	160	Using the /FORMAT Command	170
Running a Main Batch Program with DB2 Access.	161	Running Transaction Programs as IMS MPPs	171
Running Main Batch Program with DL/I Access	161	IMS Commands	171
Running a Main Batch Program with DB2 and DL/I Access	162	Keyboard Key Operation	171
Recovery and Restart for Batch Programs	164	DBCS Data on a Non-DBCS Terminal	171
		Error Reporting	172
		Responding to IMS Error Messages	172
		Running Batch Programs as MPPs	173
		Running a Main Program under IMS BMP	173
		Examples of Runtime JCL for IMS BMP Programs	174
		Running a Main Batch Program as an IMS BMP Program	174
		Running a Main Batch Program as an IMS BMP Program with DB2 Access	175
		Recovery and Restart for IMS BMP Programs	176
Chapter 18. Preparing and Running Generated Programs in IMS/VS and IMS BMP	165		
Modifying the IMS System Definition Parameters	165	Chapter 19. Moving Prepared Programs to Other Systems from MVS Systems	177
Defining an Interactive Program	166	Moving Prepared Programs To Another MVS System	177
Defining Parameters for a Batch Program as an MPP	167	Maintaining Backup Copies of Production Libraries	178
Defining Parameters for a Batch-Oriented BMP Program	167		
Defining Parameters for a Transaction-Oriented BMP Program	168		

Chapter 12. Output of Program Generation on MVS Systems

This chapter provides an overview of the output files from program generation. The output files are transferred to the MVS host system and become members of a partitioned data set (PDS). You use these PDS members to prepare the programs for production the same way you would prepare COBOL programs you might write directly. Preparation steps include running translators, precompilers and compilers, performing link-edits, and defining control tables for the target runtime environment.

List of Program Preparation Steps after Program Generation

VisualAge Generator Server supports program preparation and installation in the MVS environments using the generated preparation JCL and the cataloged procedures shipped with VisualAge Generator Server. You must perform the steps listed in Table 16 before you can run your program in an MVS target environment.

Table 16. Preparation Steps for MVS Environments

Preparation Step	Environment
Transfer from workstation to the host	All
Database 2* (DB2*) precompile	DB2 use only
CICS translation	CICS only
COBOL compile	All
Link	All
Bind	DB2 use only. This is also required if the first program in the run unit specifies either a DB2 work database or a VisualAge Generator Server Version 1 Release 1 DB2 message database.

Additionally, for CICS and IMS environments, you must define your program and transactions to the environment. For CICS, you do this using the program properties table (PPT) and program control table (PCT) entries or the Resource Definition Online (RDO) PROGRAM and TRANSACTION entries. For information on CICS entries, see Chapter 15. Preparing and Running a Generated Program in MVS CICS. For IMS, you define your program and transactions through the IMS system definition. For information on IMS system definition, see Chapter 18. Preparing and Running Generated Programs in IMS/VS and IMS BMP.

For information about generating programs and generation options, refer to the *VisualAge Generator Generation Guide*.

Outputs of Generation

After you have generated a program, you have a number of objects that need to be transferred to the host system as members in a PDS. On the MVS system, these members need to be prepared before the program can be run.

The generator produces the following COBOL outputs:

- Code compatible with the level of COBOL specification that was supported by VS COBOL II for these environments:
 - MVS batch
 - MVS CICS
 - IMS BMP
 - IMS/VS
 - TSO
 - VSE batch
 - VSE CICS
 - VM CMS
 - VM batch

The VS COBOL II source code is compatible with the COBOL for MVS and VM compiler and the LE runtime library.

- Code compatible with the VisualAge COBOL specification for the OS/2 CICS environment.
- Code compatible with the ILE COBOL/400 specification for the OS/400 environment.

Note: Generated program and map group objects are environment dependent. All program or map group objects are generated for one environment and cannot be used in another environment. Table COBOL programs are system dependent, but not environment dependent. A table generated and prepared for one environment can be used in another environment on the same system without generating the table again for that environment.

Table 17 on page 115 provides information about the types of files produced by generation, including:

- Type of object produced
- Low-level qualifier of the PDS name to which the object is written

- How the member name of the object is derived
- Runtime environments for which the object is produced
- Whether production is controlled by a COBOL generation option
- Whether the object can be modified after generation is performed

Refer to the *VisualAge Generator Generation Guide* for more information on controlling and modifying generation and preparation of MVS objects.

A description of each object begins on page 117.

Table 17. Objects Generated for Programs and Transferred to an MVS Host by the VisualAge Generator Preparation Utility

File Type	PDS Low-level Qualifier	PDS Member Name and OS/2 File Name	MVS Runtime Environments	Generation Option	Modifiable
Application COBOL program	EZESRC	Application name <i>applname.CBL</i>	All	None	No
Sample runtime JCL	EZEJCLX	Application name <i>applname.JCX</i>	IMS BMP, MVS batch	/RUNFILE	Review and possible modification required
Sample runtime CLIST	EZECLST	Application name <i>applname.CLX</i>	MVS/TSO	/RUNFILE	Review and possible modification required
BIND command	EZEBIND	Application name <i>applname.BND</i>	All	/PREPFILE, /BIND	Yes

Table 18. Objects Generated for Tables and Transferred to an MVS Host by the VisualAge Generator Preparation Utility

File Type	PDS Low-level Qualifier	PDS Member Name and OS/2 File Name	MVS Runtime Environments	Generation Option	Modifiable
Table COBOL program	EZESRC	Table name <i>tblname.CBL</i>	All	/GENTABLES	No

Table 19. Objects Generated for Map Groups and Transferred to an MVS Host by the VisualAge Generator Preparation Utility

File Type	PDS Low-level Qualifier	PDS Member Name and OS/2 File Name	MVS Runtime Environments	Generation Option	Modifiable
Online print services program - (See note 1)	EZESRC	Map group name <i>mapgname.CBL</i>	MVS CICS, MVS/TSO	/GENMAPS /GENHELPMAPS	No
MFS print services program - (See note 1)	EZESRC	Map group name <i>mapgname.CBL</i>	IMS, IMS BMP (if /MSP=ALL or /MSP=MFS)	/MSP option for IMS BMP /GENMAPS /GENHELPMAPS	No
Batch print services program - (See note 1)	EZESRC	Map group name with suffix P1 <i>mapgnameP1.CBL</i>	IMS BMP (non-MFS Maps), MVS batch	/MSP /GENMAPS /GENHELPMAPS	No
Map group format module - (See note 2)	EZEFOBJ	Map group name with suffix FM <i>mapgnameFM.FMT</i>	MVS CICS, MVS/TSO, IMS, IMS BMP	/GENMAPS /GENHELPMAPS	No
MFS control blocks	EZEMFS	Map group name <i>mapgname.MFS</i>	IMS, IMS BMP (if /MSP=ALL or /MSP=MFS)	/MSP option for IMS BMP /GENMAPS /GENHELPMAPS	No
COBOL copybook for MFS MID/MOD layout (See note 2)	EZECOPY	Map group name <i>mapgname.CPY</i>	IMS, IMS BMP (if /MSP=ALL or /MSP=MFS)	/MSP option for IMS BMP /GENMAPS /GENHELPMAPS	No

Table 20. Objects Generated for All Member Types (Programs, Tables, Map Groups) and Transferred to an MVS Host by the VisualAge Generator Preparation Utility

File Type	PDS Low-level Qualifier	PDS Member Name and OS/2 File Name	MVS Runtime Environments	Generation Option	Modifiable
Listing File	(See note 4)	Member specified when generation was requested <i>mbrname.LST</i>	All	/LISTING /LISTING ONERROR	None
Parts Control File (See note 3)	EZEPCTL	Member specified when generation was requested <i>mbrname.PCF</i>	All	None	No

Table 20. Objects Generated for All Member Types (Programs, Tables, Map Groups) and Transferred to an MVS Host by the VisualAge Generator Preparation Utility (continued)

File Type	PDS Low-level Qualifier	PDS Member Name and OS/2 File Name	MVS Runtime Environments	Generation Option	Modifiable
Preparation Command File	(See note 4)	Part specified when generation was requested <i>mbrname.PRP</i>	All	/PREPFILE	No
Preparation JCL	EZEJCLP	Application name <i>applname.JCP</i>	All	/PREPFILE	Yes
CICS Program and Transaction Definitions (See note 5)	EZEPPT (program macro definitions), EZEPCP (transaction macro definitions)	Part specified when generation was requested <i>applname.PPT</i> or <i>applname.PCT</i>	MVS CICS	/CICSENTRIES	Review and possible modification required

Notes:

1. This object is produced only if the map group contains print maps.
2. This object is produced only if the map group contains terminal maps.
3. One parts control file is created for a GENERATE subcommand. The parts control file has the same name as the part specified on the GENERATE subcommand. If any map groups or tables are generated with a program, the parts control file for the program lists all parts generated for the program and all parts generated for the associated map groups and tables.
4. These files are not transferred to the host. The OS/2 file name is the same as the name of the part specified on the GENERATE subcommand.
5. If you specify the /CICSENTRIES=RDO generation option, the PROGRAM entries are placed in *applname.PPT* and the TRANSACTION entries are placed in *applname.PCT*.

Objects Generated for Programs

Application COBOL Program

The generated program is a COBOL program that contains the following:

- Program control logic
- Logic for program processes, statement groups, and I/O operations
- Data for both the program and program control

The program control logic performs the following functions for a program, as needed:

- Initialization
- Cleanup at end of program

- Error reporting
- Segmentation support, including environment saving and restoration
- Transfer of control

Sample Runtime JCL

The generator produces sample JCL for running programs in the IMS BMP and MVS batch environments when the /RUNFILE generation option is specified during program generation. Each person using the JCL must provide a JOB statement.

The JCL is produced from model JCL templates that can be modified to enforce customer data set naming conventions. Refer to the *VisualAge Generator Generation Guide* for information about modifying the sample templates.

The JCL might not be complete and should be reviewed and modified if necessary before being used. For example, the JCL for the generated program does not contain any DD statements for data sets used by other programs that can be started by CALL, DXFR, or XFER statements. Comments in the JCL indicate where DD statements for these programs need to be added. To build the final JCL needed to run a set of programs as a run unit, you should edit the program JCL and include the DD statements for invoked programs with the JCL for the first main program. You might need to add DD statements for files that are specified during run time with the EZEDEST or EZEDESTP special function words.

Sample Runtime CLIST

The generator produces a sample CLIST for running a program in the MVS/TSO environment.

The CLIST is produced from model CLIST templates that can be modified to enforce customer data set naming conventions. Refer to the *VisualAge Generator Generation Guide* for information about modifying the sample templates.

The CLIST might not be complete and should be reviewed and modified if necessary before being used. For example, the CLIST for the generated program does not contain file allocations for programs that are started using EZEAPP on the XFER or DXFR statements. Comments in the CLIST indicate where the allocation commands for these programs need to be added. To build the final CLIST needed to run a set of programs in a run unit, you should edit the CLIST for the first main program and include commands to run the runtime CLISTs for any programs transferred-to using EZEAPP on a DXFR or XFER statement.

BIND Commands

BIND commands are required for an SQL program. You can define these BIND commands prior to generation in workstation files with the same name as the program in the directory defined by the /BIND generation option. VisualAge Generator builds a default BIND command if you do not code your own DB2 BIND commands.

The default BIND command is produced from model BIND command templates that can be modified to enforce customer standards. Refer to the *VisualAge Generator Generation Guide* for information about modifying the templates.

Objects Generated for Tables

Table COBOL Program

The table program is a COBOL program that contains the table contents defined in program working storage. This object is produced when you specify the /GENTABLES generation option. This allows tables to be generated independently of programs when the contents of a table need to be changed.

Objects Generated for Map Groups

Online Print Services Program

The online print services program is a COBOL program that performs print I/O, output formatting, and SET map CLEAR and EMPTY functions for a generated online MVS/TSO or MVS CICS program. This object is produced when you specify the /GENMAPS or /GENHELPMAPS generation options during program generation.

Message Format Service (MFS) Print Services Program

In IMS/VS and IMS BMP environments, MFS is used to build the terminal and printer data stream. The MFS print services program is a COBOL program that provides functions similar to the online print services program, and in addition builds the MVS message output descriptor (MOD) for the map being printed. This object is produced when you specify the /GENMAPS or the /GENHELPMAPS generation options.

Batch Print Services Program

The batch print services program is a COBOL program that formats data for line printers and writes the data to either the printer output file (directly to the printer or a QSAM file) or to a generalized sequential access method (GSAM) file. This program is used with programs that run in the IMS BMP or

MVS batch environments. This object is produced when you specify the /GENMAPS or the /GENHELPMAPS generation options.

Map Group Format Module

The map group format module is a generated structure that describes the map layout for terminal maps in the map group. The generator builds the structure as an MVS object module for the MVS/TSO, MVS CICS, and IMS/VS environments. This object is produced when you specify the /GENMAPS or the /GENHELPMAPS generation options.

MFS Control Blocks

MFS control blocks are generated for the IMS/VS and IMS BMP target environments using MFS utility control statements. The following control blocks are generated:

- The (device output format and device input format, DOF/DIF) control blocks describing the messages that MFS sends to or receives from the display device.

One format (FMT) is generated for each map group. One device (DEV) statement is generated for each 3270-type device that has a screen size large enough to contain one of the display maps defined in the map group. A DEV statement is generated for each printer device that has a line-length long enough to support any of the printer maps defined in the map group. If all maps contain double-byte character set (DBCS) or mixed fields, only DBCS devices are supported. Single-byte maps are supported on both non-DBCS and DBCS devices.

Individual map formats are described as device pages (DPAGE) within the format.

- The (message input descriptor, MID) control block describing how MFS formats the input message so that the generated program can process it.
A MID is generated for each map group. The input message associated with an individual map is defined as a logical page (LPAGE).
- The (message output descriptor, MOD) control block describing the output message that the generated program sends to MFS.
A MOD is generated for each map group. The output message associated with an individual map is defined as a logical page.

Refer to the IMS documentation for your system for additional information about the MFS control blocks.

COBOL Copybook for MFS MID/MOD Layout

For terminal maps created as MFS source, a COBOL copybook is generated that defines the layout of the MFS message input and message output descriptors. This copybook can be used by non-VisualAge Generator programs that perform deferred message switching to or from VisualAge Generator programs. One copybook is generated for each map group.

Objects Generated for All Part Types

The following lists the objects generated for part types of program, table, and map group.

Listing File

A source listing of the part being generated and its associates is produced if one of the print options (/LISTING or /LISTINGONERROR) is specified during generation. The listing includes any messages that were issued and options that were in effect during generation.

Parts Control File

The parts control file is a list of all objects produced by the generator the last time generation was performed for the program, table, or map group with the same name as the parts control file part. This list provides a record of generated objects.

Preparation Command File

A preparation command file is generated when the /PREPFILE generation option is specified. The preparation command file produced by a GENERATE subcommand contains logic to prepare the generated parts. If the generated program includes tables and maps, the logic for preparing them is contained in the preparation command file.

The preparation command file runs on the workstation and contains commands used to transfer the source files and preparation JCL file to the target host system. One preparation command file is created for each GENERATE subcommand that was issued.

Preparation JCL for MVS Environments

The generator produces JCL statements to precompile, translate, compile, link, and bind generated programs. The JCL produced depends on the nature of the program:

- SQL programs require DB2 precompile and BIND steps.
- CICS programs require CICS translation and must link with CICS modules.

- A program that issues DL/I calls must link with the appropriate DL/I modules.
- MFS control blocks require an MFS assemble and link.
- Map group format modules require a link-edit step.
- COBOL programs must compile and link.

The JCL is produced from model JCL templates that you can modify to enforce data set naming conventions or preparation procedures. Refer to the *VisualAge Generator Generation Guide* for more information.

The preparation command file on the workstation transfers the preparation JCL to the MVS host system and submits it to the MVS internal reader.

You can define BIND commands for specific programs prior to generation in workstation files with the same name as the program in the directory defined by the /BIND generation option. The file extension is BDC.

You can define linkage editor control statements for specific programs prior to generation in workstation files with the same name as the program in the directory specified in the /LINKEDIT generation option. The file extension is LEC.

MVS CICS Program and Transaction Definitions

CICS program and transaction definitions for the MVS CICS environment are generated as specified in the /CICSENTRIES option. The definitions might need to be modified before using in the CICS system definition.

Chapter 13. Preparing Generated Programs on MVS Systems

This chapter describes preparation procedures that are common to all runtime environments. These preparation procedures include the following steps:

- Running the preparation job
- Analyzing the preparation listing
- Preparing and running a DB2 program

Allocating Preparation Data Sets

VisualAge Generator Developer builds a set of objects necessary for running programs in MVS environments. VisualAge Generator Developer also builds and runs a preparation command file on the OS/2 system that transfers the generated objects to an MVS system and submits a preparation job (one of the generated objects) to the internal reader to complete the preparation process.

The transferred objects are stored in partitioned data sets. You allocate the required data sets using the ELACUSER CLIST shipped in data set ELA.V1R2M0.ELACLST. This CLIST was customized at product installation to set keyword default values to settings appropriate for your environment.

For you to use this CLIST, data set ELA.V1R2M0.ELACLST **must** be placed before data set ELA.V1R2M0.SELACLST in the SYSPROC concatenation list. Make sure that every COBOL generation user has the required data sets allocated for **every** environment in which the product will be used.

The following keyword parameters within CLIST ELACUSER may either be customized within the CLIST or overridden when executing the CLIST:

Keyword	Possible Values
----------------	------------------------

IMSVS	
--------------	--

- Y = allocate user data sets for this environment
- N = do not allocate user data sets for this environment

MVSCICS	
----------------	--

- Y = allocate user data sets for this environment
- N = do not allocate user data sets for this environment

MVSBATCH	
-----------------	--

- Y = allocate user data sets for this environment

- N = do not allocate user data sets for this environment

IMSBMP

- Y = allocate user data sets for this environment
- N = do not allocate user data sets for this environment

TSO

- Y = allocate user data sets for this environment
- N = do not allocate user data sets for this environment

VOL

vvvvvv = serial number

UNIT

uuuuu = valid unit name

HLQ

hhhhhhh = high-level qualifier for user data sets

CLST

- FB = allocate a fixed blocked CLIST library
- VB = allocate a variable blocked CLIST library

DB2

- Y = DB2 databases will be used with this product
- N = DB2 databases will not be used with this product

CBLK

cccccc = CLIST data set block size

LBLK

lllll = load library data set block size

An example of the command syntax to execute the CLIST is as follows:

```
ex 'ela.v1r2m0.elaclst(elacuser) imsvs(n) mvscics(y) mvsbatch(y)
imsbmp(n) tso(n) vol(at1235) unit(sysda) hlq(tsouid) db2(y)'
```

Table 21 describes the data sets that are allocated. The DD name in the table is the DD name used to allocate the data sets in the preparation job.

Table 21. Program Preparation User Data Set Information

DD Name	Data Set Name	Description	DCB Information	Target Environment
DBRMLIB	cghlq.env.DBRMLIB	Database request module library for DB2 programs	DSORG=PO, RECFM=FB, BLKSIZE=6160, LRECL=80	All MVS, if DB2 used
EZEBIND	cghlq.env.EZEBIND	Bind commands	DSORG=PO, RECFM=FB, BLKSIZE=6160, LRECL=80	All MVS, if DB2 used

Table 21. Program Preparation User Data Set Information (continued)

DD Name	Data Set Name	Description	DCB Information	Target Environment
EZECLST	cghlq.env.EZECLST	MVS/TSO CLIST library	<ul style="list-style-type: none"> • DSORG=PO, RECFM=FB, BLKSIZE=6160, LRECL=80 • DSORG=PO, RECFM=VB, BLKSIZE=3120, LRECL=255 	TSO
EZECOPY	cghlq.env.EZECOPY	Generated message input descriptor (MID) and message output descriptor (MOD) layout copybooks	DSORG=PO, RECFM=FB, BLKSIZE=6160, LRECL=80	IMSVS, IMSBMP
EZEJCLP	cghlq.env.EZEJCLP	Program preparation job stream, PDS	DSORG=PO, RECFM=FB, BLKSIZE=6160, LRECL=80	All MVS
EZEJCLX	cghlq.env.EZEJCLX	Batch program runtime job stream	DSORG=PO, RECFM=FB, BLKSIZE=6160, LRECL=80	MVSBATCH, IMSBMP
EZEMFS	cghlq.env.EZEMFS	Generated message format services control block source	DSORG=PO, RECFM=FB, BLKSIZE=6160, LRECL=80	IMSVS, IMSBMP
EZEFOBJ	cghlq.env.EZEFOBJ	Map group format object modules	DSORG=PO, RECFM=FB, BLKSIZE=3120, LRECL=80	TSO, MVSCICS, IMSVS, IMSBMP
EZEPCTL	cghlq.env.EZEPCTL	Parts control file	DSORG=PO, RECFM=FB, BLKSIZE=6160, LRECL=80	All
EZESRC	cghlq.env.EZESRC	COBOL source library	DSORG=PO, RECFM=FB, BLKSIZE=6160, LRECL=80	All

Table 21. Program Preparation User Data Set Information (continued)

DD Name	Data Set Name	Description	DCB Information	Target Environment
SYSLIN	cghlq.env.OBJECT	Object library	DSORG=PO, RECFM=U, BLKSIZE=6144, LRECL=0	All MVS
SYSMOD	cghlq.env.LOAD	Load library	DSORG=PO, RECFM=U, BLKSIZE=6144, LRECL=0	All MVS
EZEPCT	cghlq.env.EZEPCT	CICS PCT entries or RDO TRANSACTION entries	DSORG=PO, RECFM=FB, BLKSIZE=6160, LRECL=80	MVSCICS
EZEPPT	cghlq.env.EZEPPT	CICS PPT entries or RDO PROGRAM entries	DSORG=PO, RECFM=FB, BLKSIZE=6160, LRECL=80	MVSCICS

Note: where:

cghlq Specifies the high-level qualifier

env Specifies one of the following environments: IMSVS, MVSCICS, MVS BATCH, IMSBMP, or TSO

Running the Preparation JCL

VisualAge Generator submits the preparation JCL file built by the generator to the MVS internal reader. If the preparation job ends with an error that can be corrected without requiring the program to be generated again, you can use the VisualAge Generator PREPARE subcommand to submit the preparation JCL file to the MVS internal reader again.

Analyzing the Preparation Listing

If you have errors in the generated preparation JCL, check the generation listing for error messages related to the generation of JCL. Errors can occur if the JCL templates used to create the preparation JCL are not modified correctly, if the templates do not match changes you made to the VisualAge Generator JCL procedures, or if you incorrectly modified the VisualAge Generator JCL procedures. Refer to the *VisualAge Generator Generation Guide* for information about modifying the templates.

For error codes you might encounter during the precompile or bind steps, see “Chapter 26. Common Messages during Preparation for MVS Systems” on page 245 for more information.

Analyzing the DB2 Precompile Step

The precompile step is successful if the return code is 4 or less.

You should not receive any messages from the DB2 precompiler. If you receive an error message, see “DB2 Precompiler and Bind Messages” on page 246 for a list of messages that might be received from the DB2 precompiler. If you cannot determine the cause of the error from this manual or from the appropriate DB2 messages manual, have your system administrator contact the IBM Support Center.

Analyzing the CICS Translator Step

The translator step is successful if the return code is 4 or less.

You should not receive any messages from the CICS translator. If you receive messages, have your system administrator contact the IBM Support Center.

Analyzing the Compile Step

The compile step is successful if the return code is 4 or less.

Several common compiler warning messages are possible if you do any of the following:

- Use SQL row records
- Use the OPTIMIZE compile option
- Assign or move a data item to itself
- Use /MATH=COBOL

These common messages are listed in “COBOL Compilation Messages” on page 248. If you encounter COBOL messages other than those listed in that section, check the EZEPRINT listing from generation for any error messages. Correct these errors and then generate and prepare the program again. If there are no errors in EZEPRINT, give the COBOL compile listing that contains the error and the associated COBOL code to your system administrator and have the administrator contact the IBM Support Center.

Analyzing the Link-Edit Step

This section discusses how to resolve link-edit errors.

Link-Edit Errors Related to Missing Modules

Some modules from products such as CICS, DB2, and VisualAge Generator Server must be included in the load module. A small set of functions are linked with each generated COBOL program. The VisualAge Generator Server modules are included with one or more linkage editor statements in the following form:

```
INCLUDE SELALMD(modulename)
```

Other VisualAge Generator Server modules are linked into one of several modules and are loaded dynamically at run time. These modules can be preloaded or placed in the link pack area.

If you use dynamic linkage, the link-edit step should always end with a return code of 0. If you receive a nonzero return code, check for IEW0132 messages in the listing. IEW0132 messages indicate that a module cannot be found. Check that the link-edit step includes a DD statement that points to the library where the modules are stored.

If you use static linkage and you defined a linkage editor control statement (LEC) file on the workstation specifically for the generated program, a separate relink step (using procedure ELARLINK) is included in the preparation JCL. The initial link-edit step and the relink step end with return code 8 until all modules are available in the program load library, at which time the relink step should give a return code of 0, indicating that the program is ready to run. Refer to the VisualAge Generator Client/Server Communications Guide document for additional information on specifying the linkage editor control statements for statically linked modules.

Other Link-Edit Errors

If you receive a nonzero return code for some reason other than a module not found in the initial link-edit step, refer to the linkage editor manual for an explanation of the messages. If the relink step is not successful using the linkage editor control statements you provided in the EZELINK library, review the information about specifying linkage editor control statements for static linkage in the VisualAge Generator Client/Server Communications Guide document, and refer to the error information in the linkage editor manual to determine the cause of the error.

Analyzing the BIND Step

The BIND step is only needed when a DB2 program is prepared and a new database request module (DBRM) is created. If you use SQL, the preparation JCL includes a BIND step. The BIND step runs even if the link-edit step is not successful. The BIND step ends with a condition code 8 if any of the programs included in the BIND have not yet been generated.

Unless there is an error because some of the (DBRMs) are not available, you should not receive any messages from the BIND step. If you receive an error message, do the following:

1. Review the information on writing BIND commands in the *VisualAge Generator Generation Guide*.
2. See “DB2 Precompiler and Bind Messages” on page 246 for a list of messages that might be received from the BIND step.

Preparing to Run a DB2 Program

This section describes the tasks that need to be done to enable program users to run a DB2 program, such as granting authority and defining synonyms for DB2 tables.

Granting Authority to Program Users

When you have finished preparing your generated program, you need to grant authority to the program users so they can run the program using the DB2 plan. You can do this by using the following GRANT command:

```
GRANT EXECUTE ON PLAN myplan TO user1, user2, user3;
```

If the RETAIN keyword is specified on the BIND command for a plan, you need to grant authority one time for each plan user. If RETAIN is not specified, you must grant authority each time the plan is bound.

Defining Synonyms for DB2 Tables

Depending on how you specified the table when you defined your SQL row records in the program, you might need to create a synonym for the table before you can run the program:

- If you specified a fully qualified table name when you defined your SQL row record, then you do not need to do anything special when you do the BIND.
- If you did not specify a fully qualified table name when you defined your SQL row record, then you need to do one of the following:
 - If you use static SQL, then you, as the plan owner, need to create a synonym with the CREATE SYNONYM command to point to the final DB2 table that is to be used when the program is run. You need to do this before anyone else tries to run the program.
 - If you use dynamic SQL, then, before running the program, each user of the program needs to create a synonym with the CREATE SYNONYM command to point to the final DB2 table that is to be used.

For additional information on using unqualified table names and synonyms, refer to the *VisualAge Generator Design Guide*.

If the program was generated to use the DB2 implementation of the VisualAge Generator Server work database (or the message database from IBM Cross System Product Runtime Services Version 1 Release 1), the plan owner must also have created ELAWORK or ELAMSG synonyms to point to the appropriate work or message database tables.

Using LE Runtime Options for MVS/TSO and MVS Batch

Runtime options cannot be passed to generated COBOL programs through a PARM statement in the JCL or the program PARM parameter because the COBOL program is not the primary entry in the module. The primary entry in the module is ELARMAIN, an assembler program that is required for doing transfer of control through the use of the OS XCTL macro.

An alternate way to pass run unit options to COBOL programs is presented here for you to consider.

The LE programming guide explains how to create a program defaults module, CEEUOPT, that can be linked to a program to set run unit options for that program. CEEUOPT is then used to resolve the weak external reference when the generated COBOL program is linked.

Several of these modules can be created with different run unit options specified in each module. Each of the modules uses the CSECT name CEEUOPT. Then when you want to use a particular set of run unit options, you can link your program with the module that has the particular set of run unit options you want to use, for example:

```
Module CEEUOPTA was created to use run unit options A1, A2, and A3.  
Module CEEUOPTB was created to use run unit options B1, B2, and B3.  
APPLA needs run unit options A1, A2, and A3.  
APPLB needs run unit options B1, B2, and B3.
```

then

```
APPLA is linked with CEEUOPTA.  
APPLB is linked with CEEUOPTB.
```

In the following examples, OPTLIB is the DD statement pointing to the library where the CEEUOPT modules are stored and APPLIB points to the library containing the generated COBOL programs.

The linkage editor control statements for APPLA should look like this:

```
//L.SYSIN DD *
INCLUDE OPTLIB(CEEUOPTA)
INCLUDE APPLIB(APPLA)
NAME APPLA(R)
```

The linkage editor control statement for APPLB should look like this:

```
//L.SYSIN DD *
INCLUDE OPTLIB(CEEUOPTB)
INCLUDE APPLIB(APPLB)
NAME APPLB(R)
```

There are several ways you can cause the CEEUOPT version that you need to be linked with your program:

- Modify the preparation JCL after generation to add additional DD statements and linkage editor control statements.
- Modify the templates to add the additional DD statements and linkage editor control statements. Control which CEEUOPT module is used by using a symbolic parameter and the /SYMPARM generation option.
- Link the programs again after preparation to link in the CEEUOPT module you want to use.

Figure 11 on page 132 shows sample JCL in which a program needs to be linked again to change the run unit options. The JCL uses the ELAL PROC. The actual linkage editor control statements and SYSLIB used can vary for different target systems and program types.

```

//*****
//* ELAL - COBOL LINK-EDIT
//*
//* LICENSED MATERIALS - PROPERTY OF IBM
//* 5648-B02 (C) COPYRIGHT IBM CORP. 1994, 1998
//* SEE COPYRIGHT INSTRUCTIONS
//*
//* STATUS: VERSION 1, RELEASE 2, LEVEL 0
//*
//*****
//*
//ELAL PROC CGHLQ='USER',
// COBCICS='CEE.SCEECICS',
// COBLIB='CEE.SCEELKD',
// ELA='ELA.VIR2M0',
// DFHLOAD='DFH.SDFHLOAD',
// ENV='IMSVS',
// MBR=TEMPNAME,
// RESLIB='IMSESA.RESLIB',
// RGN=512K,
// SOUT='*',
// WSPC=500
//*
//* PARAMETERS:
//* CGHLQ = COBOL GENERATION USER DATA SET HIGH LEVEL QUALIFIER
//* COBCICS = LE CICS RUN TIME LIBRARY
//* COBLIB = LE RUN TIME LIBRARY
//* ELA = VISUALAGE GENERATOR SERVER HIGH LEVEL QUALIFIER
//* DFHLOAD = CICS LOAD LIBRARY
//* ENV = COBOL GENERATION USER DATA SET ENVIRONMENT QUALIFIER
//* (SHOULD BE EQUAL TO GENERATION TARGET ENVIRONMENT)
//* MBR = SOURCE NAME
//* RESLIB = IMS RESLIB LIBRARY
//* RGN = REGION SIZE
//* SOUT = SYSOUT ASSIGNMENT
//* WSPC = PRIMARY AND SECONDARY SPACE ALLOCATION
//*
//*****
//*
//* TAILORING:
//*
//* THIS PROC IS INITIALLY SET UP FOR CICS ENVIRONMENTS THAT ALSO
//* HAVE IMS SUPPORT.
//*
//* A) REMOVAL OF CICS SUPPORT:
//* 1. REMOVE 'COBCICS' FROM THE PARAMETERS AND ALL LINES THAT USE
//* THAT PARAMETER.
//* 2. REMOVE 'DFHLOAD' FROM THE PARAMETERS AND ALL LINES THAT USE
//* THAT PARAMETER.
//*
//* B) REMOVAL OF IMS SUPPORT:
//* 1. REMOVE 'RESLIB' FROM THE PARAMETERS AND ALL LINES THAT USE
//* THAT PARAMETER.
//*

```

Figure 11. Using the COBOL Linkage Editor to Select Run Unit Options (Part 1 of 2)

```

//*****
//*          LINK-EDIT THE COBOL PROGRAM
//*****
//*
//L          EXEC PGM=IEWL,REGION=&RGN,
//          PARM='RENT,REUS,LIST,XREF,MAP,AMODE(31),RMODE(ANY)'
//SYSLIB DD  DISP=SHR,DSN=&ELA..SELALMD
//          DD  DISP=SHR,DSN=&DFHLOAD
//          DD  DISP=SHR,DSN=&COBCICS
//          DD  DISP=SHR,DSN=&COBLIB
//          DD  DISP=SHR,DSN=&RESLIB
//SELALMD DD  DISP=SHR,DSN=&ELA..SELALMD
//SYSLIN DD  DISP=SHR,DSN=&CGHLQ..&ENV..OBJECT(&MBR)
//          DD  DDNAME=SYSIN
//SYSLMOD DD  DISP=SHR,DSN=&CGHLQ..&ENV..LOAD(&MBR)
//SYSPRINT DD  SYSOUT=&SOUT,DCB=BLKSIZE=13300
//SYSUDUMP DD  SYSOUT=&SOUT,DCB=BLKSIZE=13300
//SYSUT1 DD  SPACE=(1024,(&WSPC,&WSPC)),UNIT=VIO
//*          PEND          REMOVE * FOR USE AS INSTREAM PROCEDURE

```

Figure 11. Using the COBOL Linkage Editor to Select Run Unit Options (Part 2 of 2)

Notes:

1. These values are customized during your installation process.
2. These are overrides to the PROC.
3. ddname SOPTLIB should point to the data set that contains your different CEEUOPTx modules.

Chapter 14. Creating or Modifying Runtime JCL or CLISTs on MVS Systems

This chapter contains the information you need to modify the sample runtime JCL or CLIST created during program generation. If you specified /RUNFILE at generation, the generator created JCL (for the MVS batch or IMS BMP environments) or an MVS/TSO CLIST (for the MVS/TSO environment) for running the generated program. You might need to modify the runtime JCL or CLISTs for the following reasons:

- The generator does not include DD statements in the JCL to allocate data sets accessed by programs called by or transferred to from the generated program.
- The generator does not include ALLOCATE commands in the CLIST to allocate data sets accessed by programs transferred to using EZEAPP from the generated program.
- The generator does not include DD statements or ALLOCATE commands to allocate data sets accessed when an allocation moves an alternate ddname to EZEDEST or EZEDESTP. DD statements or ALLOCATE commands are not required if a program moves a data set name to EZEDEST or EZEDESTP.
- The generator does not create any recovery or restart JCL or CLIST procedures.
- The sample JCL or CLIST is based on the initial program in the run unit.

You need to ensure that the load libraries containing the initial program and any dynamically invoked programs are included in the STEPLIB concatenation unless you are using methods to put the load modules in memory. This includes program modules that are dynamically called, programs that are transferred to by an XFER or DXFR statement, and print services programs, map group format modules, and VisualAge Generator tables.

Tailoring JCL or CLIST Templates before Generation

VisualAge Generator creates sample runtime JCL for batch programs being generated for the MVS batch or IMS BMP environments, and sample runtime CLISTs for the MVS/TSO environment. The sample runtime JCL or CLIST is based on templates supplied with VisualAge Generator. The default templates shipped with VisualAge Generator are installed in the TEMPLATE subdirectory of the VisualAge Generator product directory.

Your system administrator can tailor the JCL or CLISTS that are generated by changing the JCL or CLIST templates prior to generation. If you always need to make the same changes to the generated JCL or CLISTS, tailoring the templates can help reduce the number of manual changes required for the generated JCL and CLISTS.

Some of the reasons to tailor the JCL or CLIST templates include the following:

- Implementing your installation's data set naming conventions
- Adding DD statements to the STEPLIB concatenation
- Specifying a different DB2 subsystem
- Routing output to a system printer instead of a data set for CLISTS

Table 22 shows the relationship between the JCL templates used, the target environments, and the types of databases being used by the program. It also refers you to sample JCL.

Table 22. Runtime JCL Templates Based on Environment and Databases

JCL Template	Environment	Databases	Sample JCL
EFK0MEBE	MVS batch	No DL/I or DB2	Figure 20 on page 160
EFK0MEBD	MVS batch	With DB2, no DL/I	Figure 21 on page 161
EFK0MEBC	MVS batch	With DL/I, no DB2	Figure 22 on page 162
EFK0MEBB	MVS batch	With DL/I and DB2	Figure 23 on page 163
EFK0MEIB	IMS BMP	With no DB2	Figure 28 on page 175
EFK0MEIA	IMS BMP	With DB2	Figure 29 on page 176
EFK0MEBA	MVS batch or IMS BMP, called program	Any combination	DD statements

Table 23 shows the JCL templates that serve as models for DD statement generation for program-dependent files and databases.

Table 23. Model DD Statement for Program-Dependent Files and Databases

JCL Template	Contents
EFK0MSDI	QSAM input file
EFK0MSDO	QSAM output file

Table 23. Model DD Statement for Program-Dependent Files and Databases (continued)

JCL Template	Contents
EFK0MVSI	VSAM input file
EFK0MVSO	VSAM output file
EFK0MGSI	GSAM input file
EFK0MGSO	GSAM output file
EFK0MDLI	DL/I Database DD statement for MVS batch
EFK0MIMS	GSAM IMS data sets for IMS BMP
EFK0MCAL	Comment indicating where to insert DD statements for known transferred-to and called programs
EFK0MEZA	Comment indicating where to insert DD statements for programs transferred-to using EZEAPP
EFK0MEZD	Comment indicating where to insert DD statements for data sets using EZEDEST or EZEDESTP

Refer to the *VisualAge Generator Generation Guide* for more information and for examples of tailoring JCL and CLIST templates.

Modifying Runtime JCL

The sample runtime JCL for main batch programs contains EXEC statements to run a program or a cataloged procedure. The JCL for main batch programs does not include a JOB statement or the DD statements for data sets accessed by called or transferred-to programs. Before you use the JCL to run the program, you must do the following:

- Add a JOB statement.
- Insert missing DD statements as required. Comments in the generated JCL indicate where to insert the DD statements.

The sample runtime JCL for called programs contains only the DD statements that are required for the program.

After generation, add the DD statements for any files required by called or transferred-to programs (including those named with EZEAPP) to the sample JCL for the main program. In addition, you must add DD statements for any files accessed by moving an alternate ddname to EZEDEST or EZEDESTP. You do not need to add DD statements for files that you access by moving a data set name to EZEDEST or EZEDESTP. You can also customize the sample

runtime JCL with respect to specific data set name assignments, DCB information, output file space allocations, additional steps, and other relevant data.

The type of runtime JCL generated for a main batch program varies based on the types of databases used by the main program, as shown in Table 22 on page 136 . The generated runtime JCL does not consider the types of databases accessed by called or transferred-to programs. If the main program does not use relational databases, but it calls or transfers to programs that use relational databases, you must modify the runtime JCL for the main program. This situation does not occur for DL/I because the main program must have a PSB defined even if it does not access DL/I databases.

For example, consider the following situation:

- Program A is a main batch program that does not use relational databases.
- Program B is main program that accesses relational databases.
- Programs A and B are generated for the MVS batch environment.
- Program A transfers to program B using a DXFR statement.

Because program A does not use DB2, the JCL generated for program A is for a main batch program without DB2 access (as shown in Figure 20 on page 160). This JCL will not run correctly because program B requires DB2 to run. However, the JCL generated for program B is for an MVS batch job with DB2 access (as shown in Figure 21 on page 161). The runtime JCL for program B can serve as a starting point for creating the JCL required to run program A. The following changes are required to the runtime JCL for program B:

- Change RUN PROG(APPLB) to RUN PROG(APPLA).
- Add any DD statements for files required by program A or other programs in the job step.

If program B is a called program and program A calls B rather than using a DXFR statement, the runtime JCL for program B consists only of DD statements. In this situation, you need to create your own program JCL. Any one of the following can serve as a starting point for the JCL:

- The runtime JCL for another main program that accesses relational databases.
- The JCL template for the appropriate combination of DL/I and DB2
- The examples listed in Table 22 on page 136 for the appropriate combination of DL/I and DB2

You can avoid the modification just described if you include an SQL process option in the initial main program.

If you get a JCL error for the runtime JCL, check the generation listing for the programs involved for any error messages related to JCL generation. In addition, ensure the tailoring that was done for the JCL templates is correct. Also check any changes you made when you customized the sample runtime JCL.

Modifying Runtime CLISTS

VisualAge Generator creates a sample runtime CLIST for main transaction and main batch programs being generated for the MVS/TSO environment. You cannot call the generated CLIST from a REXX procedure.

The sample runtime CLIST is based on CLIST templates supplied with VisualAge Generator. The sample CLIST for called programs contains only the ALLOCATE commands that are required for files used by the program.

The runtime CLIST actually generated is based on the resources required by the program and the specific program data set names. The examples referred to in this section were generated from the CLIST templates shipped with VisualAge Generator. Your generated CLIST might be different if your administrator has tailored the CLIST templates to reflect the standards of your organization. Refer to the *VisualAge Generator Generation Guide* for information on modifying the CLIST templates.

The sample runtime CLIST for main transaction or main batch programs contains the basic MVS/TSO commands to run the program. It also contains commands to run the runtime CLIST for programs or non-VisualAge Generator programs that are called or explicitly transferred to by a DXFR or an XFER statement. This results in the allocation of any data sets needed by the called or transferred-to program. However, the generated CLIST does not include commands to run the CLISTS for programs that were transferred-to using EZEAPP on the DXFR or XFER statement. If you use EZEAPP, and the transferred-to program uses additional data sets, you must insert the commands to run the transferred-to program's CLIST before you use the CLIST for the first program in the run unit.

After generation, add the commands to run the CLISTS for any programs that are transferred to using EZEAPP. In addition, if you do not use CLISTS to perform the data set allocations for non-VisualAge Generator programs that are called or explicitly transferred to using a DXFR or an XFER statement, you need to remove the statements that run the CLISTS for these programs. You must also add ALLOCATE (ALLOC) commands for files accessed by these non-VisualAge Generator programs. In addition, you must add ALLOCATE (ALLOC) commands for any files accessed by moving an alternate ddname to EZEDEST or EZEDESTP. You do not need to add ALLOC commands for files

that you access by moving a data set name to EZEDEST or EZEDESTP. You can also customize the sample runtime CLIST with respect to specific data set name assignments, DCB information, output file space allocations, additional steps, and other relevant data. You can also change the routing for VisualAge Generator Server and COBOL output data sets (ELAPRINT, ELASNAP, EZEPRINT, SYSABOUT, and SYSOUT) to a system spool rather than a user data set.

The type of runtime CLIST generated for a main transaction or a main batch program varies based on the types of databases used by the main program, as shown in Table 24 on page 141. The generated runtime CLIST does not consider the types of databases accessed by called or transferred-to programs. If the main program does not use relational databases, but it calls or transfers to programs that use relational databases, you must modify the runtime CLIST for the main program. This situation does not occur for DL/I because the main program must have a PSB defined even if it does not access DL/I databases.

For example, consider the following situation:

- Program A is a main transaction program that does not use relational databases.
- Program B is a main transaction program that accesses relational databases.
- Programs A and B are generated for the MVS/TSO environment.
- Program A transfers to program B using a DXFR statement.

Because program A does not use DB2, the CLIST generated for program A is for an MVS/TSO program without DB2 access (as shown in Figure 17 on page 154). This CLIST will not run correctly because program B requires DB2 to run. However, the CLIST generated for program B is for an MVS/TSO program with DB2 access (as shown in Figure 18 on page 155). The runtime CLIST for program B can serve as a starting point for creating the CLIST required to run program A. The following changes are required to the runtime CLIST for program B:

- Change RUN PROG(APPLB) to RUN PROG(APPLA).
- Add any ALLOC commands for files required by program A.
- Replace program B's data set allocations by program A's data set allocations. Add a command to run the CLIST for program B so the data sets needed by program B will also be allocated. This way, if program B is generated again and uses different data sets, program A's CLIST will not be affected.

If program B is a called program and program A calls B rather than using a DXFR statement, the runtime CLIST for program B consists only of the allocation commands and control information to prevent duplicate allocations.

In this situation, you need to create your own program CLIST. Any one of the following can serve as a starting point for the CLIST:

- The runtime CLIST for another main program that accesses relational databases
- The CLIST template for the appropriate database use
- The examples listed in Table 24 for the appropriate database use

You can avoid the modification just described if you include an SQL process option in the initial main program.

If you get an error while using the runtime CLIST, check the generation listing for the programs involved for any error messages related to CLIST generation. In addition, ensure the tailoring that was done for the CLIST templates is correct. Also check any changes you made when you customized the sample runtime CLIST.

Table 24 shows the relationship between the model CLIST templates and the types of databases being used by the programs.

Table 24. Runtime CLIST Models Based on Databases

Model CLIST	Databases	Sample CLIST
EFK0METD	No DL/I or DB2	Figure 17 on page 154
EFK0METC	With DB2, no DL/I	Figure 18 on page 155
EFK0METB	With DL/I, no DB2	Figure 19 on page 157
EFK0META	Allocations for called programs	Allocation commands only

Table 25 shows the CLIST templates that serve as models for allocation commands for program dependent files and databases.

Table 25. Model CLISTs for File Allocation

Model CLIST	Contents	CLIST Procedure Called
EFK0MTSI	Allocate QSAM input file	
EFK0MTSO	Allocate QSAM output file	ELATALC
EFK0MTVI	Allocate VSAM input file	
EFK0MTVO	Allocate VSAM output file	
EFK0MTDL	DL/I database allocation statement for MVS/TSO	
EFK0MTCL	Allocate files for known transferred-to or called programs	CLIST for transferred-to or called program

Table 25. Model CLISTs for File Allocation (continued)

Model CLIST	Contents	CLIST Procedure Called
EFK0MTEA	Comment indicating where to insert commands to run the CLISTs for programs transferred-to using EZEAPP	
EFK0MTED	Comment indicating where to insert ALLOCATE commands for data sets using EZEDEST or EZEDESTP	

Chapter 15. Preparing and Running a Generated Program in MVS CICS

This chapter describes the unique steps required to prepare and run a generated COBOL program in an MVS CICS environment:

- Modifying CICS resource tables
- Modifying CICS startup JCL
- Making new modules available
- Making programs resident
- Running programs

For general information on preparing your program for the runtime environment, see Chapter 13. Preparing Generated Programs on MVS Systems.

Modifying CICS Resource Tables

The CICS environment uses tables to identify startup parameters, transactions, programs, files, databases, transient data destinations, and system locations for proper operation. You must add to or modify these tables to correctly identify all objects to be used in the new or changed program. The CICS tables are compiled as assembler programs and stored in a runtime library. Some tables can also be maintained through an online facility as described in the resource definition online manual for your version of CICS. CICS/ESA requires that the online facility be used in place of processing program table (PPT) and program control table (PCT) entries.

The following examples show sample entries for the various CICS table types needed for runtime support of generated programs. Refer to the MVS CICS operations guide for instructions on assembling and link-editing CICS tables. Refer to the section on developing client/server programs in the *VisualAge Generator Design Guide* for special considerations related to remote files and client/server processing.

You can either write your own PPT, PCT, or RDO program and transaction entries or use the ones generated by VisualAge Generator. You must handle DCT, FCT, and RCT entries yourself.

Program Entries (PPT)

The entries in the PPT define programs to CICS. The VisualAge Generator generation process creates programs that must be defined in the PPT or as a resource definition online (RDO) PROGRAM entry.

An entry is required for each VisualAge Generator table, main program, dynamically called program, map group format module, and online print services program. You can request that sample PPT or RDO entries be generated for you by specifying the /CICSENTRIES option at generation. If you did not request generation of sample entries, you can determine what programs were generated by looking at the parts control file for the generated program, map group, or table. Each entry with a part type of APPL, MAPGROUP, MAPFM, or TABLE requires a corresponding entry in the PPT. The following figures show example PPT entries:

```
DFHPPT TYPE=ENTRY,           X
      PROGRAM=applname,      X
      PGMLANG=COBOL,         X
      PGMSTAT=ENABLED,       X
      RELOAD=NO,             X
      RES=NO                  1
```

Figure 12. PPT Entry for Generated Program

1 RES=YES might provide better performance for frequently used programs.

```
DFHPPT TYPE=ENTRY,           X
      PROGRAM=tblname,       X
      PGMLANG=COBOL,         X
      PGMSTAT=ENABLED,       X
      RELOAD=NO,             X
      RES=NO                  1
```

Figure 13. PPT Entry for Generated Table

1 RES=YES might provide better performance for frequently used single user tables.

```
DFHPPT TYPE=ENTRY,           X
      PROGRAM=mapgrp,        X
      PGMLANG=COBOL,         X
      PGMSTAT=ENABLED,       X
      RELOAD=NO,             X
      RES=NO                  1
```

Figure 14. PPT Entry for Online Print Services Program

1 RES=YES might provide better performance for frequently used map groups.

DFHPPT TYPE=ENTRY,	X	
PROGRAM=mapgrpFM,	X	
PGMLANG=ASSEMBLER,	X	1
PGMSTAT=ENABLED,	X	
RELOAD=NO,	X	
RES=NO		2

Figure 15. PPT Entry for Map Group Format Module

- 1 ASSEMBLER is required for the FM module.
- 2 RES=YES might provide better performance for frequently used map groups.

If you specify /CICSENTRIES=RDO, CICS RDO DEFINE PROGRAM commands are generated for you from templates for each program that requires an RDO PROGRAM entry. The preparation command file created during generation copies the RDO command files to the MVS library specified at generation. Refer to the *VisualAge Generator Generation Guide* for information on how to modify templates for generating RDO commands.

The following examples show how to define the same entries using the RDO CEDA transaction DEFINE PROGRAM command. Use the following command for defining the program, table, and online print services COBOL programs where *progname* is the program, table, or map group name. The L(COBOL) parameter should be changed to L(LE370) if you compile the program using COBOL/370.

```
CEDA DEF PROG(progname) L(COBOL) REL(NO) RES(NO) S(ENABLED)
```

Use the following command for the map group format module, where *mapgrpFM* is the map group format module.

```
CEDA DEF PROG(mapgrpFM) L(ASSEMBLER) REL(NO) RES(NO) S(ENABLED)
```

The values shown for REL, RES, and S keywords are the default values and can be omitted from the command. RES(YES) might provide better performance for frequently used programs.

Transaction Entries (PCT)

The PCT contains the control information used by CICS for identifying and initializing a transaction. This table is required by CICS to verify incoming requests to start transactions, and to supply information about the transaction such as the transaction priority, the security key, and the length of the transaction work area (TWA).

A CICS PCT entry or an RDO TRANSACTION entry is required for each transaction code used to start a VisualAge Generator generated main program. This includes transactions with the following attributes:

- Started from a clear CICS screen
- Used as an EZESEGTR name
- Used as the target of an XFER or CALL CREATX statement
- Transferred to by a non-VisualAge Generator program
- Started by other CICS facilities
- Used as the name of the segmented transaction ID specified at generation

Programs that are invoked through a DXFR or CALL statement do not require a PCT entry.

Sample entries are generated for the main transaction and segmented transaction identifiers if you specify the /CICSENTRIES generation option.

The following figure shows a sample PCT entry for a generated program:

```

DFHPCT TYPE=ENTRY,           X
      PROGRAM=applname,      X  1
      TRANSID=tran,         X  2
      DTB=YES,              X  3
      DUMP=NO,              X  4
      RESTART=NO,          X  5
      TWASIZE=1024         X  6
  
```

Figure 16. PCT Entry for Generated Program

- 1 Name of program to be started for this transaction. If the transaction being defined is one that is used as the EZESEGTR value for a segmented CONVERSE, specify the name of the program that did the CONVERSE for best performance.
- 2 Assign a unique transaction ID.
- 3 Uncommitted changes are backed out if an abend occurs. If the program updates recoverable resources, you must specify DTB=YES. If the program is inquiry-only, specifying DTB=NO might provide better performance.
- 4 VisualAge Generator Server contains an exit that is activated when your transaction ends abnormally. The exit is enabled by a CICS HANDLE ABEND statement at initialization time. The exit issues CICS DUMP commands if an abend occurs unless the diagnostic control setting for the transaction is set to bypass dumps. Specifying DUMP=YES causes CICS to issue a transaction dump when transactions end abnormally. If you specify DUMP=YES, use the

VisualAge Generator Server diagnostic controller utility to set the dump option for abends to "no dump." See "Chapter 21. Diagnostic Control Options" on page 189 for more information about the VisualAge Generator Server diagnostic controller utility.

- 5 Specify RESTART=YES if you intend to use the transaction restart facility and the EZEDLRST special word.
- 6 The size for the CICS Transaction Work Area (TWA) must be at least 1024 bytes. If you are running a non-VisualAge Generator program together with VisualAge Generator programs and the program requires a TWA, you must increase the TWASIZE parameter by the amount your non-VisualAge Generator program requires and specify the /TWAOFF option when generating the VisualAge Generator programs. Refer to the *VisualAge Generator Generation Guide* for information about the /TWAOFF generation option.

The following example shows how to define the same transaction using the RDO CEDA transaction DEFINE TRANSACTION command. The same comments apply to the command parameters as apply to the PCT entry. The I(BACKOUT) parameter is equivalent to DTB=YES. DTB=NO cannot be specified with the DEFINE command.

```
CEDA DEF TR(tran) PROG(appName) I(BACKOUT) DU(NO) RES(NO) TW(1024)
```

Destination Control Table Entries (DCT)

A CICS destination control table (DCT) entry is required for each program file that is assigned to a transient data queue. A DCT entry is also required for destinations specified as error destination queue names using the VisualAge Generator Server diagnostic controller utility. The parameters for DCT entries depend on your destination type. There are intrapartition, extrapartition, indirect, and remote destinations. See "Using and Allocating Data Files in MVS CICS and VSE CICS" on page 69 for information about defining and managing program data files and "Defining Transient Data Queues" on page 72 for information about defining the DCT entry for the error destination queue. Refer to appropriate CICS manuals for more information on DCT entries.

File Control Table Entries (FCT)

A CICS file control table (FCT) entry is required for each program file that is specified as file type VSAM. You must identify all FCT entries that might be referenced at run time. Files can also be defined using RDO. See "Using and Allocating Data Files in MVS CICS and VSE CICS" on page 69 for more information on defining and managing program data files in the CICS environment.

Resource Control Table Entry (RCT)

If the programs running under a transaction access DB2 databases, then you must define an entry in the CICS resource control table (RCT) that associates the transaction identifier with the program plan name.

The following example shows the minimum RCT entry required:

```
DSNCRCT TYPE=ENTRY TXID=tran PLAN=plan-name
```

For more information on the other parameters you can specify when you define RCT entries, refer to the chapter on connecting the CICS attachment facility in the DB2 installation manual for your version of DB2.

Using Remote Programs, Transactions, or Files

Refer to the section on developing client/server programs in the *VisualAge Generator Design Guide* for information about the support of remote programs, transactions, or files. Also refer to the appropriate CICS manuals for information about defining remote programs, transactions, or files.

Using MVS JES Spool Files

Refer to the *VisualAge Generator Design Guide* for information about using spool files.

Modifying CICS Startup JCL

You must include the load library where your generated programs reside in the DFHRPL DD concatenation. Your system administrator included the LE runtime libraries and the VisualAge Generator Server Version 1 Release 1 load library in the DFHRPL DD concatenation when the VisualAge Generator Server Version 1 Release 2 product was installed.

The CICS startup JCL might need to be modified to add or change allocations for files used by VisualAge Generator programs. These include VSAM files and extrapartition transient data destinations.

For VSAM data sets, it is not necessary to include allocations in the startup JCL if you specify the data set name and disposition in the CICS FCT or RDO entry for the file. CICS dynamically allocates the file at open time.

Making New Modules Available in the MVS CICS Environment

After you generate a new version of a program, map group, or table, you need to make the modules available to CICS.

For programs and map groups, you can use the CICS NEWCOPY command or the VisualAge Generator Server new copy utility to cause the new copy of the program to be used the next time a load request is issued for the program. If you use the CICS NEWCOPY command for a map group, you must issue the NEWCOPY for both the online print services program and the map group format module.

For tables, you must use the VisualAge Generator Server new copy utility to cause a fresh copy of the table to be used the next time a load request is issued for the table. Do not use the CICS NEWCOPY command for tables. The VisualAge Generator Server new copy utility sets a flag indicating that the new copy of the table is to be used the next time a program loads the table contents.

For more information on the VisualAge Generator Server new copy utility, see “New Copy” on page 183.

Making Programs Resident

You can make frequently used programs or programs with high performance requirements resident to avoid the overhead of loading the programs when they are used. To aid in deciding which programs should be made resident, you can use CICS shutdown statistics to determine how often a generated program is loaded in a CICS region.

To make program and map group programs resident, specify the program as resident in the PPT or RDO entry for the program. To make table programs resident, specify the table as resident when you define the table in VisualAge Generator Developer. Refer to the MVS CICS Resource Definition (Online) manual for more information about the CICS resident option. Refer to the *VisualAge Generator Design Guide* for information about the VisualAge Generator resident option for table definition.

Running Programs under MVS CICS

Either a main transaction program or main batch program generated for MVS CICS can be started with CICS facilities. Called programs can only be started by another VisualAge Generator program or by a non-VisualAge Generator program.

Prior to starting a generated program, the program user might be required to sign on to the CICS environment. Refer to CICS documentation for information about signing on.

Starting the Transaction in MVS CICS

Any main program that is generated with a target environment of MVS CICS can be started by entering the transaction code associated with the main program from a clear screen in CICS. Any main program that is started directly in CICS, by an XFER statement from another program, or by a CREATX service routine must have a unique transaction code assigned to it. The transaction code must be defined in the PCT or with an RDO TRANSACTION entry and be associated with the first program in the run unit.

Controlling Diagnostic Information in the CICS Environment

VisualAge Generator Server provides a diagnostic controller utility for the CICS environment. This utility allows you to control the type of dump, the name of the error destination queue and journal number for error messages, and whether the transaction is disabled when a run unit error occurs. See “Chapter 21. Diagnostic Control Options” on page 189 for more information about the diagnostic controller utility.

Printing Diagnostic Messages in the CICS Environment

VisualAge Generator Server provides a way to print diagnostic messages written to a transient data queue. See “Diagnostic Message Printing Utility” on page 186 for more information.

Chapter 16. Preparing and Running a Generated Program in MVS/TSO

This chapter describes the unique steps required to prepare and run a generated COBOL program in an MVS/TSO environment:

- Making VisualAge Generator Server, COBOL, and the generated modules available
- Running main programs
- Examples of runtime CLISTS

For general information on preparing your program for the runtime environment, see Chapter 13. Preparing Generated Programs on MVS Systems. For information on modifying CLISTS, see Chapter 14. Creating or Modifying Runtime JCL or CLISTS on MVS Systems.

Making VisualAge Generator Server, COBOL, and the Generated Modules Available

If your generated program contains DL/I or SQL processes, the load library where your generated programs reside needs to have its contents accessible by one of the following methods:

- Adding the library to the STEPLIB concatenation list for your MVS/TSO logon procedure
- Using the ISPF LIBDEF ISPLLIB command
- Using the TSOLIB command provided by TSO

If your generated program does not contain DL/I or SQL processes, the MVS/TSO CALL command that is used to start the program establishes a temporary TASKLIB search chain. This TASKLIB search chain is searched when locating or loading the load module. The TASKLIB is only available during the CALL command. Therefore, if neither DL/I nor DB2 are used, and all the programs (applications, online print services programs, map group format modules, table programs, and non-VisualAge Generator programs) for the run unit are in the same load library, the above techniques are not required.

For all types of programs, the VisualAge Generator Server and the LE runtime libraries must be made available using one of the above techniques. Your system administrator might have included these libraries in the STEPLIB concatenation list when VisualAge Generator Server was installed and customized.

Running Main Programs under MVS/TSO

Either a main transaction program or a main batch program generated for MVS/TSO can be started from the MVS/TSO command line. Called programs can only be started by another VisualAge Generator program or by a non-VisualAge Generator program.

VisualAge Generator creates a sample runtime CLIST for running programs in the MVS/TSO environment. The generated CLIST has the same name as the program. If you specify the /RUNFILE generation option, a sample CLIST is created specifically for the program ID during program generation. The runtime JCL is transferred to an MVS partitioned data set (PDS) by the VisualAge Generator prepare function.

The CLIST might need to be modified to run the CLISTS for other programs that are transferred to using the EZEAPP special function word. The CLIST also might need to be modified to include data set allocation (ALLOC) commands for data sets accessed by setting the EZEDEST or EZEDESTP special function words. See “Modifying Runtime CLISTS” on page 139 for more information on modifying the sample runtime CLISTS.

If you get an error while using the runtime CLIST, check the generation listing for the programs involved for any error messages related to CLIST generation. In addition, ensure the tailoring that was done for the CLIST templates is correct. Also check any changes you made when you customized the sample runtime CLIST.

In the MVS/TSO environment, a generated program is started by a CLIST.

Prior to starting a generated program, the program user must log on to the MVS/TSO environment. Refer to the MVS/TSO documentation for information about logging on.

Starting the CLIST

If the data set containing the runtime CLIST is in the MVS/TSO SYSPROC concatenation list, you can start it by entering the following:

```
%AAAA
```

where AAAA is the name of the CLIST. Because the CLIST and program have the same name, you must enter the % for the CLIST to distinguish it from the program name.

If the CLIST data set is not in the SYSPROC concatenation list, you can start the program by entering the following:

```
EXEC 'cgh1q.env.EZECLST(AAAA)'
```

where `cgh1q.env.EZECLST` is the name of the data set where your CLIST resides, and `AAAA` is the name of the CLIST.

Examples of Runtime CLISTs for MVS/TSO

The sample CLISTs in Figure 17 on page 154 through Figure 19 on page 157 are examples of the different types of CLISTs generated by VisualAge Generator for main programs. The generated CLISTs in the following examples have these characteristics:

- The examples are based on the CLIST templates shipped with VisualAge Generator. Your actual CLIST templates might differ if your system administrator has tailored them for your organization. Refer to the *VisualAge Generator Generation Guide* for more information about tailoring CLIST templates.
- Lowercase text appears in the examples where a generic example name has been substituted for an actual program or data set name.

The logic of the generated CLISTs has the following characteristics:

- The program name is added to a global variable that contains a list of all the programs that have had their data sets allocated.
- Allocations for the output files required by VisualAge Generator Server (ELAPRINT, ELASNAP, EZEPRINT, SYSABOUT, and SYSOUT) are only performed if this is the first CLIST being run. They are allocated using a general-purpose allocation CLIST, ELATALC, which is provided in ELA.V1R2M0.ELACLST with VisualAge Generator Server.
- EZEPRINT must be a variable blocked data set.
- If the CLIST is the first one in the run unit (ALLOCATEONLY(NO)), then the program is started. If the CLIST is not the first one started (ALLOCATEONLY(YES)), then only data set allocation occurs.

Running a Main Program with No Database Access

Figure 17 on page 154 shows a CLIST generated to invoke a main transaction or main batch program that does not require database access.

```

PROC 0 ALLOCATEONLY(NO)
CONTROL NOLIST NOFLUSH MSG
GLOBAL ELAAPPS
SET NAMESTART = &SYSINDEX(app1-name,&ELAAPPS) /*IS APPL NAME IN STRING?*/
/* IF NAME FOUND, CLIST ALREADY RAN ONCE */
IF &NAMESTART > 0 THEN +
  EXIT /* EXIT IF ALREADY BEEN HERE */
ELSE +
  SET ELAAPPS = &ELAAPPS..app1-name /* ADD APPLICATION NAME TO */
  /* LIST, SEPARATED BY PERIOD */

/*****/
/* ERROR EXIT
/*****/
ERROR DO
  WRITE
  WRITE SYSPCMD = &SYSPCMD
  WRITE LASTCC = &LASTCC
  WRITE
  EXIT
END

IF &ALLOCATEONLY NE YES THEN DO /* EXIT IF ONLY ALLOCATION REQUEST */
EXEC 'ELA.V1R2M0.ELACLST(ELATALC)' +
  '&SYSUID..ELAPRINT ELAPRINT SYSDA FBA 0 1330 TRACKS 2 1 OLD'
EXEC 'ELA.V1R2M0.ELACLST(ELATALC)' +
  '&SYSUID..ELASNAP ELASNAP SYSDA VBA 0 4096 CYLINDERS 2 1 OLD'
EXEC 'ELA.V1R2M0.ELACLST(ELATALC)' +
  '&SYSUID..EZEPRINT EZEPRINT SYSDA VBA 0 4096 TRACKS 5 2 OLD'
EXEC 'ELA.V1R2M0.ELACLST(ELATALC)' +
  '&SYSUID..SYSABOUT SYSABOUT SYSDA FB 0 0 TRACKS 5 2 OLD'
EXEC 'ELA.V1R2M0.ELACLST(ELATALC)' +
  '&SYSUID..SYSOUT SYSOUT SYSDA FB 0 0 TRACKS 5 2 OLD'
END
/*****/
/* FILE ALLOCATIONS FOR THIS APPLICATION, TRANSFERRED-TO APPLICATIONS
/* AND CALLED APPLICATIONS.
/*****/
/* ALLOCATE FILES FOR TRANSFERRED-TO OR CALLED APPLICATION */
EXEC 'cgh1q.env.EZECLST(transferred-to-app1-name)' 'ALLOCATEONLY(YES)'
IF &ALLOCATEONLY EQ YES THEN EXIT /* EXIT IF ONLY ALLOCATION REQ.*/

CALL 'cgh1q.env.LOAD(app1-name)'
EXIT

```

Figure 17. CLIST for Main Program without DB2 or DL/I Access

Running a Main Program with DB2 Access

Figure 18 on page 155 shows a CLIST generated to invoke a main transaction or main batch program that uses DB2.

```

PROC 0 ALLOCATEONLY(NO)
CONTROL NOLIST NOFLUSH MSG
GLOBAL ELAAPPS
SET NAMESTART = &SYSINDEX(app1-name,&ELAAPPS) /*IS APPL NAME IN STRING?*/
/* IF NAME FOUND, CLIST ALREADY RAN ONCE */
IF &NAMESTART > 0 THEN +
  EXIT /* EXIT IF ALREADY BEEN HERE */
ELSE +
  SET ELAAPPS = &ELAAPPS..app1-name /* ADD APPLICATION NAME TO */
  /* LIST, SEPARATED BY PERIOD */

/*****/
/* ERROR EXIT
/*****/
ERROR DO
  WRITE
  WRITE SYSPCMD = &SYSPCMD
  WRITE LASTCC = &LASTCC
  WRITE
  EXIT
END

IF &ALLOCATEONLY NE YES THEN DO /* EXIT IF ONLY ALLOCATION REQUEST */
EXEC 'ELA.V1R2M0.ELACLST(ELATALC)' +
  '&SYSUID..ELAPRINT ELAPRINT SYSDA FBA 0 1330 TRACKS 2 1 OLD'
EXEC 'ELA.V1R2M0.ELACLST(ELATALC)' +
  '&SYSUID..ELASNAP ELASNAP SYSDA VBA 0 4096 CYLINDERS 2 1 OLD'
EXEC 'ELA.V1R2M0.ELACLST(ELATALC)' +
  '&SYSUID..EZEPRINT EZEPRINT SYSDA VBA 0 4096 TRACKS 5 2 OLD'
EXEC 'ELA.V1R2M0.ELACLST(ELATALC)' +
  '&SYSUID..SYSABOUT SYSABOUT SYSDA FB 0 0 TRACKS 5 2 OLD'
EXEC 'ELA.V1R2M0.ELACLST(ELATALC)' +
  '&SYSUID..SYSOUT SYSOUT SYSDA FB 0 0 TRACKS 5 2 OLD'
END
/*****/
/* FILE ALLOCATIONS FOR THIS APPLICATION, TRANSFERRED-TO APPLICATIONS */
/* AND CALLED APPLICATIONS. */
/*****/
/* ALLOCATE FILES FOR TRANSFERRED-TO OR CALLED APPLICATION */
EXEC 'cgh1q.env.EZECLST(transferred-to-app1-name)' 'ALLOCATEONLY(YES)'
IF &ALLOCATEONLY EQ YES THEN EXIT /* EXIT IF ONLY ALLOCATION REQ.*/

ERROR OFF
DSN SYSTEM(ssid)
DATA
  RUN PROG(app1-name) PLAN(app1-name) -
  LIB('cgh1q.env.LOAD')
  END
ENDDATA

```

Figure 18. CLIST for Main Program with DB2 Access (Part 1 of 2)

```
SET MYCC = &LASTCC
IF &MYCC ^ = 0 THEN +
DO
  WRITE
  WRITE SYSPCMD = &SYSPCMD
  WRITE LASTCC = &MYCC
  WRITE
  EXIT
END
EXIT
```

Figure 18. CLIST for Main Program with DB2 Access (Part 2 of 2)

Running a Main Program with DL/I Access

Figure 19 on page 157 shows a CLIST generated to start a main transaction or main batch program that uses DL/I.

```

PROC 0 ALLOCATEONLY(NO)
CONTROL NOLIST NOFLUSH MSG
GLOBAL ELAAPPS
SET NAMESTART = &SYSINDEX(app1-name,&ELAAPPS) /*IS APPL NAME IN STRING?*/
/* IF NAME FOUND, CLIST ALREADY RAN ONCE */
IF &NAMESTART > 0 THEN +
  EXIT /* EXIT IF ALREADY BEEN HERE */
ELSE +
  SET ELAAPPS = &ELAAPPS..app1-name /* ADD APPLICATION NAME TO */
  /* LIST, SEPARATED BY PERIOD */

/*****/
/* ERROR EXIT
/*****/
ERROR DO
  WRITE
  WRITE SYSPCMD = &SYSPCMD
  WRITE LASTCC = &LASTCC
  WRITE
  EXIT
END

IF &ALLOCATEONLY NE YES THEN DO /* EXIT IF ONLY ALLOCATION REQUEST */
EXEC 'ELA.V1R2M0.ELACLST(ELATALC)' +
  '&SYSUID..ELAPRINT ELAPRINT SYSDA FBA 0 1330 TRACKS 2 1 OLD'
EXEC 'ELA.V1R2M0.ELACLST(ELATALC)' +
  '&SYSUID..ELASNAP ELASNAP SYSDA VBA 0 4096 CYLINDERS 2 1 OLD'
EXEC 'ELA.V1R2M0.ELACLST(ELATALC)' +
  '&SYSUID..EZEPRINT EZEPRINT SYSDA VBA 0 4096 TRACKS 5 2 OLD'
EXEC 'ELA.V1R2M0.ELACLST(ELATALC)' +
  '&SYSUID..SYSABOUT SYSABOUT SYSDA FB 0 0 TRACKS 5 2 OLD'
EXEC 'ELA.V1R2M0.ELACLST(ELATALC)' +
  '&SYSUID..SYSOUT SYSOUT SYSDA FB 0 0 TRACKS 5 2 OLD'
EXEC 'ELA.V1R2M0.ELACLST(ELATALC)' +
  '&SYSUID..IEFRDER IEFRDER SYSDA VB 1916 1920 CYLINDERS 1 1 OLD'
ALLOC FILE(IMS) +
  DATASET('IMSVS.DBDLIB' +
    'IMSVS.PSBLIB') +
  SHR REUSE
ALLOC FILE(DFSRESLB) +
  DATASET('IMSVS.RESLIB') +
  SHR REUSE
ALLOC FILE(DFSVSAMP) +
  DATASET('ELA.V1R2M0.ELASAMP(ELAVSAMP)') +
  SHR REUSE
END

```

Figure 19. CLIST for a Main Program with DL/I Access (Part 1 of 2)

```

/*****/
/* FILE ALLOCATIONS FOR THIS APPLICATION, TRANSFERRED-TO APPLICATIONS
/* AND CALLED APPLICATIONS. ALSO ALLOCATIONS FOR DL/I DATABASES.
/*****/
/* ALLOCATE FILES FOR TRANSFERRED-TO OR CALLED APPLICATION */
EXEC 'cghlq.env.EZECLST(transferred-to-appl-name)' 'ALLOCATEONLY(YES)'
IF &ALLOCATEONLY EQ YES THEN EXIT /* EXIT IF ONLY ALLOCATION REQ.*/

CALL 'IMSVS.RESLIB(DFSRR00)' 'DLI,appl-name,ims-psb-name,,,,,,,,,Y,N,,Y,'
EXIT

```

Figure 19. CLIST for a Main Program with DL/I Access (Part 2 of 2)

Chapter 17. Preparing and Running Generated Programs in MVS Batch

This chapter describes the unique steps required to prepare a generated COBOL program to run in an MVS batch environment:

- Running main programs
- Examples of runtime JCL
- Recovery and restart

For general information on preparing your program for the runtime environment, see Chapter 13. *Preparing Generated Programs on MVS Systems*. For information on modifying the JCL, see Chapter 14. *Creating or Modifying Runtime JCL or CLISTs on MVS Systems*.

Running Main Programs under MVS Batch

A main batch program generated for the MVS batch environment can be started by submitting JCL. Called programs can only be started by another VisualAge Generator program or by a non-VisualAge Generator program.

VisualAge Generator creates sample runtime JCL for running programs in the MVS batch environment. The generated JCL has same name as the program. If you specify the /RUNFILE generation option, a sample JCL is created specifically for the program during program generation. The runtime JCL is transferred to an MVS partitioned data set (PDS) by the VisualAge Generator prepare function.

The JCL might need to be modified to add data sets required by called or transferred-to programs. You also need to modify the JCL to add any data sets that are dynamically allocated with the EZEDEST or EZEDESTP special function words. See “Chapter 14. Creating or Modifying Runtime JCL or CLISTs on MVS Systems” on page 135 for more information on modifying the sample runtime JCL.

If you get a JCL error for the runtime JCL, check the generation listing for the programs involved for any error messages related to JCL generation. In addition, ensure the tailoring that was done for the JCL templates is correct. Also check any changes you made when you customized the sample runtime JCL.

The following sections show JCL for different MVS batch programs.

Examples of Runtime JCL for MVS Batch Programs

The generated JCL in the following examples has these characteristics:

- The examples are based on the JCL templates shipped with VisualAge Generator. Your actual JCL templates might differ if your system administrator has tailored them for your organization. Refer to the *VisualAge Generator Generation Guide* for more information about tailoring JCL templates.
- Lowercase text appears in the examples where a generic example name has been substituted for an actual program or data set name.
- EZEPRINT is always routed to SYSOUT=*

If you route EZEPRINT to a data set, you must use the following DCB attributes:

- LRECL=137, BLKSIZE=141, RECFM=VBA if the map group does not contain any DBCS maps
- LRECL=654, BLKSIZE=658, RECFM=VBA if the map group contains any DBCS maps

You cannot use map groups that do not have any DBCS maps with map groups that do have DBCS maps in a same job step.

Running a Main Batch Program with No Database Access

Figure 20 shows the JCL used to start a main batch program.

```
//jobname JOB .....MSGCLASS=A
//stepnam EXEC PGM=appl-name
//STEPLIB DD DSN=CEE.SCEERUN,DISP=SHR
// DD DSN=ELA.V1R2M0.SELALMD,DISP=SHR
// DD DSN=cghlq.env.LOAD,DISP=SHR
//ELAPRINT DD SYSOUT=*,DCB=(RECFM=FBA,BLKSIZE=1330)
//ELASNAP DD SYSOUT=*,DCB=(RECFM=VBA,BLKSIZE=4096)
//EZEPRINT DD SYSOUT=*,DCB=(RECFM=VBA,BLKSIZE=4096)
//SYSABOUT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//* Application specific DD statements
//file-name-1 DD .....
//file-name-n DD .....
```

Figure 20. JCL for Main Batch Program Run as MVS Batch without DB2 or DL/I Access

Running a Main Batch Program with DB2 Access

Figure 21 shows the JCL used to start a main batch program that gains access to DB2 resources. The JCL must run the MVS/TSO terminal monitor program to run the generated program.

```
//jobname JOB USER=userid,.....
//stepname EXEC PGM=IKJEFT01,DYNAMNBR=20,REGION=4M
//STEPLIB DD DSN=DSN.SDSNLOAD,DISP=SHR
// DD DSN=CEE.SCEERUN,DISP=SHR
// DD DSN=ELA.VIR2M0.SELALMD,DISP=SHR
// DD DSN=cghlq.env.LOAD,DISP=SHR
//ELAPRINT DD SYSOUT=*,DCB=(RECFM=FBA,BLKSIZE=1330)
//ELASNAP DD SYSOUT=*,DCB=(RECFM=VBA,BLKSIZE=4096)
//EZEPRINT DD SYSOUT=*,DCB=(RECFM=VBA,BLKSIZE=4096)
//SYSABOUT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSTSIN DD *
DSN SYSTEM (ssid)
RUN PROG (app1-name) PLAN (plan-name) -
LIB ('cghlq.env.LOAD')
END
/*
//SYSTSPRT DD SYSOUT=*
/* Application specific DD statements
//file-name-1 DD .....
//file-name-n DD .....
```

Figure 21. JCL for Main Batch Program Run as MVS Batch with DB2 Access

Running Main Batch Program with DL/I Access

If a main batch program runs as a DL/I batch program, then all DL/I requests are handled by a private IMS region. The JCL for the step that runs the batch program must include DD statements for the IMS log if databases are opened with update intent or if the program uses the VisualAge Generator CALL AUDIT service routine. Also, a DD statement must be included for each of the data sets associated with the DL/I databases referenced in the IMS PSB. The IMS log DD statements (IEFRDER and IEFRDER2) are normally included in the DLIBATCH procedure.

VisualAge Generator uses the JCL template EFK0MDLI to build the DD statements for program databases. This template has the DD statement commented out because the high-level program database qualifiers are not collected by VisualAge Generator. You need to provide the final tailoring of these DD statements in the sample runtime JCL. Alternatively, depending on your naming conventions, your administrator might be able to modify the EFK0MDLI template so that you can use the /SYMPARM generation option to

set high-level qualifiers for databases. Refer to the *VisualAge Generator Generation Guide* for information about modifying templates and using the /SYMPARM generation option.

Figure 22 shows the sample JCL used to run a generated program as a DL/I batch program.

```
//jobname JOB .....
//stepname EXEC DLIBATCH,DBRC=Y,
//          MBR=appl-name,PSB=ims-psb-name,BKO=Y,IRLM=N
//G.STEPLIB DD
//          DD
//          DD DSN=CEE.SCEERUN,DISP=SHR
//          DD DSN=ELA.V1R2M0.SELALMD,DISP=SHR
//          DD DSN=cghlq.env.LOAD,DISP=SHR
//* DFSVSAMP IS REQUIRED IF VSAM DATABASES - REPLACE MEMBER WITH
//* ONE THAT HAS VALID BUFFER POOL SIZES FOR YOUR APPLICATION
//G.DFSVSAMP DD DSN=ELA.V1R2M0.ELASAMP(ELAVSAMP),DISP=SHR
//G.ELAPRINT DD SYSOUT=*,DCB=(RECFM=FBA,BLKSIZE=1330)
//G.ELASNAP DD SYSOUT=*,DCB=(RECFM=VBA,BLKSIZE=4096)
//G.EZEPRINT DD SYSOUT=*,DCB=(RECFM=VBA,BLKSIZE=4096)
//G.SYSABOUT DD SYSOUT=*
//G.SYSOUT DD SYSOUT=*
//* Application specific DD statements including DL/I DB DD statements
//file-name-1 DD .....
//file-name-n DD .....
```

Figure 22. JCL for Main Batch Program Run as MVS Batch with DL/I Access

Running a Main Batch Program with DB2 and DL/I Access

Figure 23 on page 163 shows the JCL that enables a program to run as a stand-alone DL/I batch processing program and to gain access to DB2 databases. Special recovery considerations are required. Refer to the DB2 documentation for your system for additional information.

The JCL for the step that runs the batch program must include DD statements for the IMS log if databases are opened with update intent or if the program uses the VisualAge Generator CALL AUDIT service routine. Also, a DD statement must be included for each of the data sets associated with the DL/I databases referenced in the IMS PSB. The IMS log DD statements (IEFRDER and IEFRDER2) are normally included in the DLIBATCH procedure.

VisualAge Generator uses the JCL template EFK0MDLI to build the DD statements for DL/I program databases. This template has the DD statement commented out because the high-level program database qualifiers are not collected by VisualAge Generator. You need to provide the final tailoring of these DD statements in the sample runtime JCL. Alternatively, depending on your naming conventions, your administrator might be able to modify the

EFK0MDLI template so that you can use the /SYMPARM generation option to set high-level qualifiers for databases. Refer to the *VisualAge Generator Generation Guide* for information about modifying templates and using the /SYMPARM generation option.

```
//jobname JOB .....
//stepname EXEC DLIBATCH,DBRC=Y,
//          MBR=DSNMTV01,PSB=ims-psb-name,BKO=Y,IRLM=N
//G.STEPLIB DD
//          DD
//          DD DSN=DSN.SDSNLOAD,DISP=SHR
//          DD DSN=CEE.SCEERUN,DISP=SHR
//          DD DSN=ELA.V1R2M0.SELALMD,DISP=SHR
//          DD DSN=cghlq.env.LOAD,DISP=SHR
/* DFSVSAMP IS REQUIRED IF VSAM DATABASES - REPLACE MEMBER WITH
/* ONE THAT HAS VALID BUFFER POOL SIZES FOR YOUR APPLICATION
//G.DFSVSAMP DD DSN=ELA.V1R2M0.ELASAMP(ELAVSAMP),DISP=SHR
/*
//G.DDOTV02 DD DSN=&&TEMP1,
//          DISP=(NEW,PASS,DELETE),
//          SPACE=(CYL,(1,1),RLSE),UNIT=SYSDA,
//          DCB=(RECFM=VB,BLKSIZE=4096,LRECL=4092)
//G.DDITV02 DD *
//          ssid,SYS1,DSNMIN10,,R,-,connection name,plan-name,appl-name
/*
//G.ELAPRINT DD SYSOUT=*,DCB=(RECFM=FBA,BLKSIZE=1330)
//G.ELASNAP DD SYSOUT=*,DCB=(RECFM=VBA,BLKSIZE=4096)
//G.EZEPRINT DD SYSOUT=*,DCB=(RECFM=VBA,BLKSIZE=4096)
//G.SYSABOUT DD SYSOUT=*
//G.SYSOUT DD SYSOUT=*
/* Application specific DD statements including DL/I DB DD statements
//file-name-1 DD .....
//file-name-2 DD .....
/*
/* Attempt to print out the DDOTV02 data set created in previous step
//stepnam2 EXEC PGM=DFSERA10,COND=EVEN
//STEPLIB DD DSN=IMSVS.RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSNNAME=&&TEMP1,
//          DISP=(OLD,DELETE)
//SYSIN DD *
CONTROL CNTL K=000,H=8000
OPTION PRINT
/*
```

Figure 23. JCL for Main Batch Program Run as MVS Batch with DB2 and DL/I Access

Recovery and Restart for Batch Programs

For MVS batch programs that use DL/I, the generated runtime JCL includes the parameter BKO=Y. If the program updates databases or files, specify BKO=Y in the runtime JCL in order to have rollback (ROLB) requests honored. If you specify BKO=N, DL/I returns status code AL for the roll-back call. VisualAge Generator Server treats the AL status code as a soft error. No error message is issued, and processing continues.

You should develop recovery procedures in the event of program or system errors. VisualAge Generator does not generate JCL to perform restart or recovery procedures.

Chapter 18. Preparing and Running Generated Programs in IMS/VS and IMS BMP

This chapter describes the steps required to prepare and run a generated COBOL program in an IMS environment:

- Modify the IMS system definition parameters
- Create the MFS control blocks
- Precompile, compile, link, and bind the generated program
- Make the new modules and MFS control blocks available to IMS
- Create or modify runtime JCL (IMS BMP only)

For general information on preparing programs for the runtime environment, see “Chapter 13. Preparing Generated Programs on MVS Systems” on page 123 . For information about modifying JCL, see “Chapter 14. Creating or Modifying Runtime JCL or CLISTS on MVS Systems” on page 135.

Modifying the IMS System Definition Parameters

The following information describes the basic IMS system definition parameters that are required to prepare VisualAge Generator programs. You should review the performance options described in the IMS documentation for your system to determine the most effective options.

An IMS TRANSACT macro is required for each transaction code used to start a VisualAge Generator generated main program in the IMS/VS environment and for each transaction-oriented BMP program. This includes the following transactions:

- Started from a clear IMS screen
- Used as an EZESEGTR name
- Used as the target of an XFER or CALL CREATX statement
- Transferred to by a non-VisualAge Generator program
- Started as the result of an ADD process option associated with the message queue
- Started by other IMS facilities

The TRANSACT macro must follow the APPLCTN macro for the IMS PSB that is to be used for the transaction.

Defining an Interactive Program

Each main transaction program must be defined as either an IMS message processing program (MPP) or a fast-path program with an associated transaction code, except when the program is started only through a DXFR statement from another program.

Figure 24 shows the system definition parameters that are required for defining an interactive VisualAge Generator program.

APPLCTN	PGMTYPE=TP,PSB=ims-psb-name.	1
TRANSACT	CODE=trancode,	X2
	INQUIRY=NO,	X3
	MODE=SNGL,	X
	MSGTYPE=(SNGLSEG,RESPONSE),	X4
	EDIT=ULC,	X5
	SPA=(size,[DASD CORE],[FIXED])	6

Figure 24. IMS System Definition for an Interactive Transaction

- 1 The IMS PSB name and the VisualAge Generator program name must match.
- 2 Multiple transactions can be associated with one program. If the program changes the value of EZESEGTTR before a CONVERSE, include a TRANSACT macro for the original transaction code and a TRANSACT macro for the EZESEGTTR value.
- 3 INQUIRY=NO is the default for IMS. If DL/I is used for the work database, INQUIRY=NO is required. The VisualAge Generator Server work database supports help maps and displays data again if an input error occurs, as well as the CONVERSE process option. Therefore, even if the program databases are inquiry only, INQUIRY=NO is necessary. If DB2 is used for the work database and the program's use of all DL/I databases is inquiry only, then INQUIRY=YES can be used.
- 4 SNGLSEG is required. Either RESPONSE or NONRESPONSE can be used with VisualAge Generator Server, depending on whether you want the keyboard to remain locked until the transaction completes. Even if NONRESPONSE mode is used, multiple simultaneous transactions from a single terminal are not supported.
- 5 Required for input in lowercase.
- 6 Include this parameter only if an IMS scratch pad area (SPA) is required. The SPA size is the length of the IMS SPA header (14 bytes) plus the length of the longest working storage record that might be received or sent during an XFER. However, if you include the position

parameter and omit the ADF parameter on the /SPA generation option, then you must add an additional byte when calculating the size. The SPA size must match the number specified for the /SPA generation option when the program is generated.

You can also include the FPATH=YES parameter on the TRANSACT macro if the program might be run in an IMS Fast Path (IFP) region. If you include FPATH=YES, be sure to include the /FASTPATH generation option when you generate the program. Refer to the IMS manuals for your system for additional information about using IFP regions.

Defining Parameters for a Batch Program as an MPP

A VisualAge Generator batch program can also run as an asynchronous MPP. For example, a VisualAge Generator batch program can be used to process the information inserted to the message queue by a CALL CREATX statement or an ADD process option in another program. This type of program differs from one that runs as a BMP in that the MPP cannot access any GSAM, indexed, or relative files, and cannot include any special restart logic. Figure 25 shows the system definition parameters required for this case.

```
APPLCTN  PGMTYPE=TP,PSB=ims-psb-name          1
TRANSACT  CODE=trancode,                       X  2
          MODE=SNGL
```

Figure 25. IMS System Definition for an Asynchronous MPP Program

- 1 The IMS PSB name and the VisualAge Generator program name must match.
- 2 Multiple transactions can be associated with one program.

You can also include the FPATH=YES parameter on the TRANSACT macro if the program might be run in an IMS Fast Path (IFP) region. If you include FPATH=YES, be sure to include the /FASTPATH generation option when you generate the program. Refer to the IMS manuals for your system for additional information about using IFP regions.

Defining Parameters for a Batch-Oriented BMP Program

If a VisualAge Generator batch program is generated to run as an IMS BMP program and it does not process an input message queue, it is a batch-oriented BMP program. Figure 26 on page 168 shows the system definition parameters required for defining a main batch program as a batch-oriented BMP program.

```
APPLCTN  PGMTYPE=BATCH,PSB=ims-psb-name
```

Figure 26. IMS System Definition for a Main Batch Program Running as a Batch-Oriented BMP Program

Defining Parameters for a Transaction-Oriented BMP Program

If a VisualAge Generator batch program is generated to run as an IMS BMP program and it processes an input message queue created by MPP programs or by other BMP programs, it is a transaction-oriented BMP program. Figure 27 shows the system definition parameters that are required to define a main batch program as a transaction-oriented BMP program.

```
APPLCTN  PGMTYPE=BATCH,PSB=ims-psb-name
TRANSACT CODE=trancode,           X   1
        MODE=SNGL,                X
        WFI                          2
```

Figure 27. IMS System Definition for a Main Batch Program Running as a Transaction-Oriented BMP Program

- 1 Multiple transactions can be associated with one program.
- 2 Wait-for-input (WFI) is optional. If it is specified, the program remains resident until the operator stops the transaction or shuts down the region.

Creating MFS Control Blocks

The message format services (MFS) control blocks can be generated using either the MFSUTL or the MFSTEST JCL procedure. These two procedures are shipped with the IMS product. When you generate the program, you specify which procedure to use with the /MFSTEST generation option. The JCL to create the MFS control blocks is part of the preparation JCL. Refer to the message format services documentation for your system for additional information about the MFS control blocks. Refer to the *VisualAge Generator Generation Guide* for information about the /MFSTEST generation option. See “Considerations for Message Format Services in IMS” on page 105 for information about other generation options (/MFSDEV, /MFSEATTR, and /MFSIGNORE) that control what is included in the MFS source.

If your program contains DBCS or mixed data, note that a long mixed constant field that results in multiple lines of MFS source might contain unpaired shift-in and shift-out characters. This occurs when the DBCS portion of the constant is split into more than one line. The MFS still works correctly.

Making New Modules Available in the IMS Environment

Whenever you install a new version of a program, MFS print services program, map group format module, or table, you need to recycle the message region.

If you generated with /MFSTEST, then the MFS control blocks were placed in the MFS test library (the TFORMAT library). To use the new version of the MFS control blocks, use the /TEST MFS command after you have signed on your IMS system and before you attempt to run a transaction that uses the new version of the maps.

If you generated with /NOMFSTEST, then the MFS control blocks were placed in the MFS staging library (FORMAT library). To use the new version of the MFS control blocks, you must do the following:

1. Run the IMS online change utility (OLCUTL) to copy the new MFS control blocks into the inactive format library.
2. Use the following IMS commands:

```
/MODIFY PREPARE FMTLIB  
/MODIFY COMMIT
```

Note: If the MFS control blocks and the map group format module do not have the same generation date and time, VisualAge Generator Server issues an error message.

Preloading Program, Print Services, and Table Modules

Preloading programs, MFS print services programs, map group format modules, and table modules that are frequently used might reduce the overhead of searching the STEPLIB, JOBLIB, link pack area, and link list. However, if modules are preloaded, they occupy virtual storage when they are not in use.

To improve response time, you might also preload modules associated with any transaction that might require better performance, even though the module itself is not frequently used.

To preload a program, MFS print services program, map group format module, or table program, have your system administrator do the following:

1. Put the module in a LNKLST library.
2. Include the module name in a preload member (DFSMPLxx, where xx is a two-character ID that you select) in IMSVS.PROCLIB.
3. Indicate in the JCL for the IMS message region that the preload member is to be included.

Refer to the IMS manuals for your system to get general information on preloading modules.

Running Programs under IMS

Prior to starting a generated program, the program user might be required to sign on to the IMS environment with a /SIGN command. Refer to the IMS documentation for information about the /SIGN command.

Starting a Main Program Directly

The simplest way for a program user to start a VisualAge Generator program is by entering the IMS transaction code from an unformatted screen. The transaction code can be up to 8 characters. It is associated with the program in the IMS system definition TRANSACT macro. The following is an example of starting a transaction:

```
MYTRANS
```

IMS requires the transaction code to be followed by at least one blank prior to pressing the ENTER key.

Starting a Main Transaction Program Using the /FORMAT Command

A program user can use the IMS /FORMAT command to display a formatted screen to start a transaction if the First Map for a program is defined with the IMS transaction code for the program as an 8-byte constant with the protect and dark attributes. The attribute byte on the map becomes the attribute byte in the generated MFS. The 8-byte constant contains the name of the IMS transaction that is started when the map is processed.

The /FORMAT command directs IMS to display a screen format; however, the command does not cause the program to be run. After the program user enters data and presses the Enter key (or a function key), the message from the terminal is sent to the generated program for processing.

The syntax of the /FORMAT command is as follows:

```
/FORMAT modname [mapname]  
      or  
/FOR modname [mapname]
```

The *modname* operand is the map group name with an *O* suffix. The *mapname* operand is required if there is more than one map in the map group. It must be the map name that was specified as the First Map for the program.

Because the transaction code must be included in the map, and a transaction code can only be associated with one program in the IMS system definition, only one program using the map can be started using the /FORMAT command.

Running Transaction Programs as IMS MPPs

Running generated programs is similar to running non-VisualAge Generator programs in the IMS MPP environment, with the following differences:

IMS Commands

The /HOLD command should be avoided. VisualAge Generator Server uses the logical terminal identifier as the key of the work database. The data in the work database is destroyed if another generated program is run from the same terminal prior to resuming the original conversation.

Keyboard Key Operation

When the Clear key is pressed in IMS, IMS clears the screen, but does not notify the program. No transaction is scheduled, so the map is not automatically displayed again. If the program is conversational, the program user can enter the IMS /HOLD command followed immediately by an IMS /RELEASE command to display the map again.

When the EOF key is pressed in the first position of a field on a map, the data is not blanked. To blank the data, the program user must enter at least one blank before pressing the EOF key. Also, the program user should not use the DELETE CHARACTER key to erase the entire field because this is equivalent to pressing the EOF key in the first position of the field.

When typing over characters in a right-justified numeric field, any intervening spaces between the new digits entered and the original digits in the field should be deleted by pressing the DELETE CHARACTER key. Alternatively, the program user can type in all digits for the new value and use the EOF key to erase any remaining digits.

DBCS Data on a Non-DBCS Terminal

If a program inadvertently attempts to display a map with DBCS or mixed data on a non-DBCS terminal or printer, the results are unpredictable. The terminal might be logged off IMS and returned to the VTAM* sign-on screen without displaying any warning or error messages. If this happens, review your use of DBCS. Also, review your values for the /MFSDEV, /MFSEATTR, and /MFSIGNORE generation options, and compare them to the IMS system definition for the terminal that had the problem.

Error Reporting

In certain error situations, VisualAge Generator Server displays its own panel to explain the error to the program user. This occurs in the following situations:

- A message needs to be displayed, but there is no EZEMSG field on the map. Map ELAM01 in map group ELAxxx, where xxx is the national language code, is used.
- An unexpected program error has occurred. Map ELAM02 and (if necessary) continuation map ELAM03 are used to display the error messages. See “Using the VisualAge Generator Server Error Panel” on page 220 for an example of ELAM02.

If an error occurs information might have been written to the message queue identified by the /ERRDEST generation option for the first program in the run unit. See “Chapter 24. IMS Diagnostic Message Print Utility” on page 203 for information on printing diagnostic errors.

Responding to IMS Error Messages

If a DFS message is displayed on your screen, make a note of the message. Then, depending on how your IMS system is set up, press either PA1 or PA2 to see if VisualAge Generator Server has queued an error map to the terminal with more information. This can happen in the following situations:

- If VisualAge Generator Server issues a ROLL call because of a run unit or catastrophic error, IMS issues the message:

```
DFS555I TRAN ttttttt ABEND S000,U0778 ; MSG IN PROCESS:
ttttttt mmmmmmmMAP                ;;;gdate gtime rdate rtime
```

Where *ttttttt* is the IMS transaction code, *mmmmmmmm* is the map name, *gdate* and *gtime* are the date and time the map group was generated, and *rdate* and *rtime* are the date and time of the abend.

The DFS555I message is also used by IMS when other abends occur, including the 1600, 1601, 1602, and 1606 abends from VisualAge Generator Server.

- If VisualAge Generator Server ends the run unit for a transaction program that was generated with /FASTPATH and is being run in an IMS fast-path region, IMS issues the message:

```
DFS2766I PROCESS FAILED
```

- If VisualAge Generator Server abnormally ends the logical unit of work (LUW) for a transaction program that was generated with /FASTPATH, IMS might issue the message:

```
DFS2082I RESPONSE MODE TRANSACTION TERMINATED WITHOUT REPLY
```

See “Chapter 25. Diagnosing Problems for VisualAge Generator Server on MVS Systems” on page 207 for information on diagnosing errors.

Running Batch Programs as MPPs

A VisualAge Generator main batch program can be generated to run in the IMS MPP environment. In this situation, IMS automatically starts the transaction whenever a message is written to the message queue associated with the transaction.

If an error occurs information might have been written to the message queue identified by the /ERRDEST generation option for the first program in the run unit. See “Chapter 24. IMS Diagnostic Message Print Utility” on page 203 for information on printing diagnostic errors.

Running a Main Program under IMS BMP

A main batch program generated for the IMS BMP environment can be started by submitting JCL. Called programs can only be started by another VisualAge Generator program or by a non-VisualAge Generator program.

VisualAge Generator creates sample runtime JCL for running programs in the IMS BMP environment. The generated JCL has the same name as the program. If you specify the /RUNFILE generation option, a sample JCL is created specifically for the program during program generation. The runtime JCL is transferred to an MVS partitioned data set (PDS) by the VisualAge Generator prepare function.

The JCL might need to be modified to add data sets required by called or transferred-to programs, or for data sets used by setting the EZEDEST or EZEDESTP special function words. You also need to modify the JCL to add any data sets that are dynamically allocated with the EZEDEST or EZEDESTP special function words. See “Chapter 14. Creating or Modifying Runtime JCL or CLISTS on MVS Systems” on page 135 for more information on modifying the sample runtime JCL.

If you get a JCL error for the runtime JCL, check the generation listing for the programs involved for any error messages related to JCL generation. In addition, ensure the tailoring that was done for the JCL templates is correct. Also check any changes you made when you customized the sample runtime JCL.

The following sections show JCL for different IMS BMP programs.

Examples of Runtime JCL for IMS BMP Programs

The generated JCL in the following examples has these characteristics:

- The examples are based on the JCL templates shipped with VisualAge Generator. Your actual JCL templates might differ if your system administrator has tailored them for your organization. Refer to the *VisualAge Generator Generation Guide* for more information about tailoring JCL templates.
- Lowercase text appears in the examples where a generic example name has been substituted for an actual program or data set name.
- EZEPRINT is always routed to SYSOUT=*

If you route EZEPRINT to a data set, you must use the following DCB attributes:

- LRECL=137, BLKSIZE=141, RECFM=VBA if the map group does not contain any DBCS maps
- LRECL=654, BLKSIZE=658, RECFM=VBA if the map group contains any DBCS maps

You cannot use map groups that do not have any DBCS maps with map groups that do have DBCS maps in a single job step.

The first library in the STEPLIB concatenation sequence must have the largest block size, or BLKSIZE=32760 can be specified on the first STEPLIB DD statement for the step.

Running a Main Batch Program as an IMS BMP Program

If a main batch program runs as a BMP program, all DL/I requests are passed to a central copy of IMS which coordinates updates to the databases across multiple BMPs and MPPs. The DD statements for the IMS log and the program databases are not required in the JCL for the BMP job step. These databases and the IMS log are allocated to the IMS control region.

Figure 28 on page 175 shows a sample set of JCL to run a generated program as a BMP program.


```

//jobname JOB .....
//stepname EXEC IMSBATCH,
//  MBR=appl-name,PSB=ims-psb-name,IN=trans-name
//G.STEPLIB DD
//      DD
//      DD DSN=CEE.SCEERUN,DISP=SHR
//      DD DSN=ELA.V1R2M0.SELALMD,DISP=SHR
//      DD DSN=cghlq.env.LOAD,DISP=SHR
//G.ELAPRINT DD SYSOUT=*,DCB=(RECFM=FBA,BLKSIZE=1330)
//G.ELASNAP DD SYSOUT=*,DCB=(RECFM=VBA,BLKSIZE=4096)
//G.EZEPRINT DD SYSOUT=*,DCB=(RECFM=VBA,BLKSIZE=4096)
//G.SYSABOUT DD SYSOUT=*
//G.SYSOUT DD SYSOUT=*
//* Application specific DD statements for files
//* No application specific DD statements for databases required
//file-name-1 DD .....
//file-name-n DD .....

```

Figure 28. JCL for Main Batch Program as an IMS BMP Program

If you run a transaction-oriented BMP program, the *trans-name* must be set to the name of the transaction for the message queue that the BMP program processes. If not, *trans-name* should be a null value. The sample runtime JCL created by VisualAge Generator defaults *trans-name* to the program name for a transaction-oriented BMP program that uses SCAN to read the message queue. The sample runtime JCL created by VisualAge Generator defaults *trans-name* to null for batch-oriented BMP programs or for transaction-oriented BMP programs that use CSPTDLI to read the message queue.

If the BMP program uses GSAM, the following DD statements are also included in the sample runtime JCL:

```

//IMS      DD DSN=IMS.PSBLIB,DISP=SHR
//          DD DSN=IMS.DBDLIB,DISP=SHR

```

These DD statements are generated from the EFK0MIMS JCL template.

Running a Main Batch Program as an IMS BMP Program with DB2 Access

Figure 29 on page 176 shows a sample set of JCL to run a generated program that accesses DB2 resources as a BMP. The DD statements for the IMS log and the DL/I program databases are not required in the JCL for the BMP job step. The DL/I databases and the IMS log are allocated to the IMS control region.

```

//jobname JOB .....
//stepname EXEC IMSBATCH,
//  MBR=appl-name,PSB=ims-psb-name,IN=trans-name
//G.STEPLIB DD
//      DD
//      DD DSN=DSN.SDSNLOAD,DISP=SHR
//      DD DSN=CEE.SCEERUN,DISP=SHR
//      DD DSN=ELA.V1R2M0.SELALMD,DISP=SHR
//      DD DSN=cghlq.env.LOAD,DISP=SHR
//G.DFSESL DD DSN=IMS.RESLIB,DISP=SHR
//      DD DSN=DSN.SDSNLOAD,DISP=SHR
//G.ELAPRINT DD SYSOUT=*,DCB=(RECFM=FBA,BLKSIZE=1330)
//G.ELASNAP DD SYSOUT=*,DCB=(RECFM=VBA,BLKSIZE=4096)
//G.EZEPRINT DD SYSOUT=*,DCB=(RECFM=VBA,BLKSIZE=4096)
//G.SYSABOUT DD SYSOUT=*
//G.SYSOUT DD SYSOUT=*
//* Application specific DD statements for files
//* No application specific DD statements for databases required
//file-name-1 DD .....
//file-name-n DD .....

```

Figure 29. JCL for Main Batch Program as an IMS BMP Program with DB2

If you run a transaction-oriented BMP program, the *trans-name* must be set to the name of the transaction for the message queue that the BMP program processes. If not, *trans-name* should be a null value. The sample runtime JCL created by VisualAge Generator defaults *trans-name* to the program name for a transaction-oriented BMP program that uses SCAN to read the message queue. The sample runtime JCL created by VisualAge Generator defaults *trans-name* to null for batch-oriented BMP programs or for transaction-oriented BMP programs that use CSPTDLI to read the message queue.

If the BMP program uses GSAM, the following DD statements are also included in the sample runtime JCL:

```

//IMS      DD DSN=IMS.PSBLIB,DISP=SHR
//         DD DSN=IMS.DBDLIB,DISP=SHR

```

These DD statements are generated from the EFK0MIMS JCL template.

Recovery and Restart for IMS BMP Programs

You should develop recovery procedures in the event of program or system error. VisualAge Generator does not generate JCL to perform restart or recovery procedures.

Chapter 19. Moving Prepared Programs to Other Systems from MVS Systems

You might need to move a prepared program from one system to another. For example you might have the compiler on one host development machine but want to run the program on several production machines.

If you use DB2, the DB2 BIND must be done on the production system.

The COBOL and VisualAge Generator Server products on the production machine must be at the same maintenance level as, or a higher level than, on the development machine.

Moving Prepared Programs To Another MVS System

If a program has been completely prepared on one system and you want to move the prepared program to another system, perform the following steps:

1. Copy the program-related parts (including the map group and table parts) in the following libraries to the production system. Library names are shown using the default naming convention used in the preparation procedures, where *cghlq* is the user or project-related high level qualifier and *env* is the runtime environment code.

Table 26. Parts to Copy

Data Set Name	Contents
<i>cghlq.env.LOAD</i>	Application, print services program, map group format and table load modules
<i>cghlq.env.DBRM</i>	DB2 database request modules (DBRMs) for SQL programs
<i>cghlq.env.BIND</i>	BIND commands for SQL programs
<i>cghlq.env.EZEMFS</i>	MFS source for IMS and IMS BMP map processing
<i>cghlq.env.EZEJCLX</i>	Runtime JCL for IMS BMP and MVS batch programs
<i>cghlq.env.EZECLST</i>	Runtime CLISTs for MVS/TSO programs

Note:

The *cghlq* variable comes from the /PROJECTID generation option. The *env* variable comes from the /SYSTEM generation option.

2. Provide your own JCL to build the plans for DB2 programs using the BIND commands from the BIND library and the DBRMs from the DBRM library. You need to edit the EZEBIND member, and change the DB2 subsystem ID to match the new system where you are moving the program.
3. Provide your own JCL to assemble the MFS control blocks for IMS and IMS BMP. It is much easier to assemble the MFS source on the production system than to try to locate the DIF/DOF and MID/MOD in the MFS format libraries. However, if you have procedures in place to move the DIF/DOF and MID/MOD to a different system, you can use them instead of moving the MFS source in the EZEMFS library.
4. Follow the procedures identified in this manual for defining programs to IMS and CICS.
5. Define files and databases used by the program on the new system.

Maintaining Backup Copies of Production Libraries

Follow your installation-defined guidelines and procedures for making backup copies of production libraries. Having backup copies of production libraries enables you to return to the prior level of a program in case of errors. The production libraries for which copies should be made are those listed in Table 26 on page 177.

You might also want to set up procedures for saving a copy of the program source in external or internal source format at the time a program is generated. Having a copy of the program source enables you to recreate production level libraries at any change level if necessary.

Part 4. Utilities for MVS Systems

Chapter 20. Using VisualAge Generator Server Utilities on MVS Systems	181
Using the CICS Utilities Menu	181
New Copy	183
Diagnostic Message Printing Utility	186
Chapter 21. Diagnostic Control Options	189
Change or View Diagnostic Control Options for a Transaction	190
Change or View Default Diagnostic Control Options	193
Chapter 22. Using the Message File Conversion Utility	195
Converting a Message File to an External Source Format Table.	195
Sample JCL for the Message File Conversion Utility	197
Chapter 23. Using the Parameter Group Utility	199
Chapter 24. IMS Diagnostic Message Print Utility	203

Chapter 20. Using VisualAge Generator Server Utilities on MVS Systems

VisualAge Generator Server provides the following utilities:

- A set of utilities in CICS to help manage the error diagnosis and control facilities of the VisualAge Generator Server runtime environment
- A batch utility for message file conversion
- A CICS utility to create and maintain parameter group files

You can access these utilities from the CICS utilities menu.

Using the CICS Utilities Menu

To access the CICS utilities do the following:

1. Log on to CICS.
2. Type ELAM on a clear screen.
3. Press Enter. When the ELAM transaction is started, a copyright panel is displayed.
4. Press Enter. The CICS Utilities Menu (Figure 30 on page 182) is displayed.

```
ELAM                               VisualAge Generator Server
                                   CICS Utilities Menu

Select one of the following utilities; then press Enter.

Action...._

_1. New Copy
_2. Diagnostic Message Printing
_3. Diagnostic Control Options

ENTER  F1=HELP  F3=EXIT
```

Figure 30. CICS Utilities Menu

Three functions are available from the CICS Utilities Menu panel (Figure 30):

New Copy

This function causes a new copy of a program, map group, or table to be used by subsequent transactions. Use the new copy function when programs, map groups, and tables are modified and generated again.

For programs and map groups, you can use the VisualAge Generator Server new copy utility or the CICS NEWCOPY command to cause the new copy of the program to be used the next time a load request is issued for the program. The VisualAge Generator Server new copy utility does a new copy for both the online print services program and the map group format module when you specify a part type of Map Group. If you use the CICS NEWCOPY command for a map group, you must issue the NEWCOPY for both the online print services program and the map group format module.

For tables, you must use the VisualAge Generator Server new copy utility to cause a fresh copy of the table to be used the next time a load request is issued for the table. Do not use the CICS NEWCOPY command for tables.

Diagnostic Message Printing

This function routes the diagnostic messages in an error destination transient data queue to a spool file for printing or subsequent processing.

Diagnostic Control Options

This function lets you view or change the diagnostic control options set for the installation or for individual transactions. The options include dump control, error message routing to a transient data queue or the CICS journal, and transaction disabling when serious problems occur.

New Copy

The VisualAge Generator Server new copy utility causes a new copy of a program, map group, or table to be used by subsequent transactions. Transactions that are in progress when this function was started continue to use the copy that was current when the transaction began. Programs must end or reach the end of a segment before the new copy is used. The VisualAge Generator Server new copy utility must be run separately for programs, map groups, and tables to replace the copy already in storage.

To gain access to the VisualAge Generator Server new copy utility, do the following:

1. Select option 1, New Copy, on the CICS Utilities Menu panel (Figure 30).
2. Press Enter.

The New Copy panel (Figure 31 on page 184) is displayed.

Note: You can also gain access to the VisualAge Generator Server new copy utility by doing the following:

1. Type ELAN on a clear screen.
2. Press Enter. When the ELAN transaction is started, a copyright panel is displayed.
3. Press Enter. The New Copy panel (Figure 31 on page 184) is displayed.

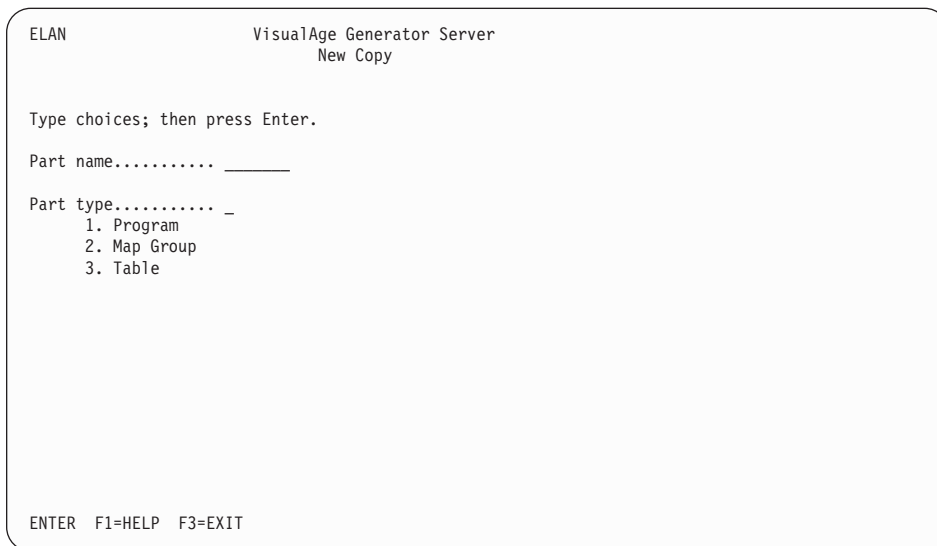


Figure 31. New Copy panel

Enter the following on the New Copy panel:

Part name

Specifies the name of the program, map group, or table to be used as a new copy in subsequent transactions

Part type

Specifies the type of part to be replaced

Note: VisualAge Generator Server does not validate the part type. You must specify the correct part type because different processing is required for programs, map groups, and tables. If you have problems in processing after using the VisualAge Generator Server new copy utility, try the VisualAge Generator Server new copy utility again to ensure you specified the part type correctly.

The type can be one of the following:

Program

This type causes the function to issue a CICS SET PROGRAM(name) NEWCOPY command to access a new copy of the program. This command does not cause a new copy for called programs that are statically linked with their caller.

Map Group

This type causes the function to issue a CICS SET PROGRAM(name) NEWCOPY command to get access to a new copy of the map group format module and the online print services program associated with the map group.

Table This type causes the utility to issue a CICS SET PROGRAM(name) NEWCOPY command to access a new copy of the table program, and sets a flag for VisualAge Generator Server indicating that the new copy of the table is to be used the next time a program loads the table contents.

If the table has been generated as a shared table, currently running transactions continue to use the old copy of the table while new transactions share the new copy of the table.

You can also access the new copy utility in batch mode. To invoke the batch new copy utility, link to program ELABNEW:

```
EXEC CICS LINK PROGRAM("ELABNEW")  
COMMAREA(passed-parms)  
LENGTH(174)
```

where the **passed-parms** record has the following structure:

Field	Length in Bytes	Type of Data	Description
NLS code	3	Character	NLS code identifying language
Part name	8	Character	name of program, map group, or table to be used as a new copy in subsequent transactions
Part type	1	Character	Type of part to be replaced: "1" Program "2" Map group "3" Table For more information, see the description for part type in the online help discussion of the new copy utility.
Return code	2	Binary	Return code from new copy
Message 1	80	Character	Message returned from new copy
Message 2	80	Character	Message returned from new copy

The following fields must be provided by the user:

- NLS code
- Part name
- Part type

The other fields are filled in by the new copy utility.

Any nonzero return code means that the new copy operation was not successful. If a nonzero value is returned in the **return code** field, check messages 1 and 2 for details indicating what error occurred.

Note: Message 2 is not always filled in. It may be blank.

Diagnostic Message Printing Utility

Diagnostic message printing allows you to route diagnostic messages in an error destination transient data queue to a JES spool file for printing.

To gain access to the diagnostic message print utility do the following:

1. Select option 2, Diagnostic Message Printing, from the CICS Utilities Menu panel (Figure 30 on page 182).
2. Press Enter.

The Diagnostic Message Printing panel (Figure 32 on page 187) is displayed.

Note: You can also access the diagnostic message print function by doing the following:

1. Type ELAU on a clear screen.
2. Press Enter. When ELAU is started, a copyright panel is displayed.
3. Press Enter. The Diagnostic Message Printing panel (Figure 32 on page 187) is displayed.

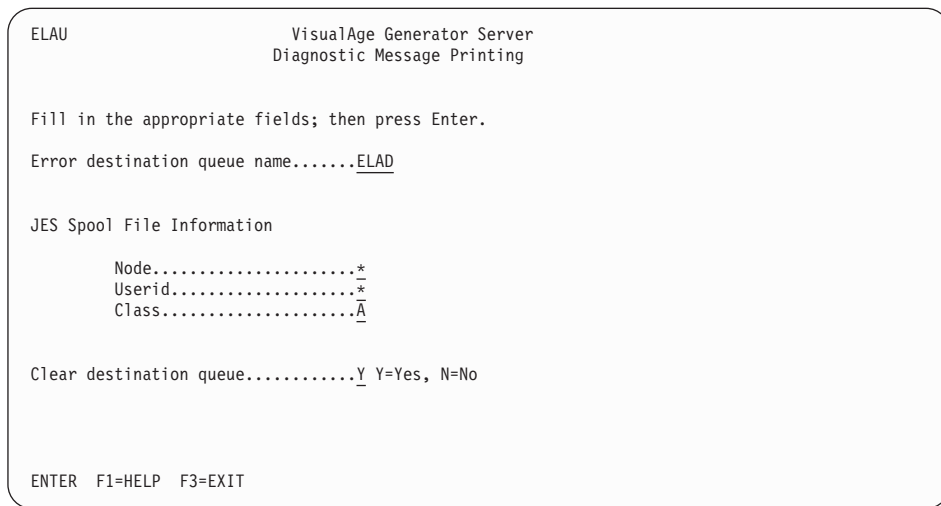


Figure 32. Diagnostic Message Printing panel

You can enter information in the following fields on the Diagnostic Message Printing panel:

Error destination queue name

This field specifies the name of an existing error destination.

Enter the 1 to 4 character DCT name of the ERRDEST error destination transient data queue. The queue name is initialized to the default error destination queue. The default is ELAD. You can either leave the messages in the queue or clear them after they have been printed.

JES Spool File Information

This field specifies the spool file where the messages are to be written. If you do not specify anything in these fields, the system uses the default values and sends the report to the local spool printer where your local MVS CICS is running.

Clear destination queue

This field specifies whether to clear the error queue of all messages after the messages are written to a spool file. The default is Y.

Chapter 21. Diagnostic Control Options

The diagnostic control options utility enables you to alter the diagnostic action options taken for a given transaction code that is assigned to a generated CICS program. If multiple transaction codes are assigned to a program, each transaction code is specified independently to the diagnostic control options utility.

You can also specify a default action to take place for transactions that are not explicitly defined to the diagnostic control options utility.

To gain access to the diagnostic control options utility, do the following:

1. Select option 3, Diagnostic Control Options, from the CICS Utilities Menu (Figure 30 on page 182).
2. Press Enter. The Diagnostic Control Options panel (Figure 33 on page 190) is displayed.

Note: You can also gain access to the diagnostic control options utility by doing the following:

1. Type ELAC on a clear screen.
2. Press Enter. When ELAC is started, a copyright panel is displayed.
3. Press Enter. The Diagnostic Control Options panel (Figure 33 on page 190) is displayed.

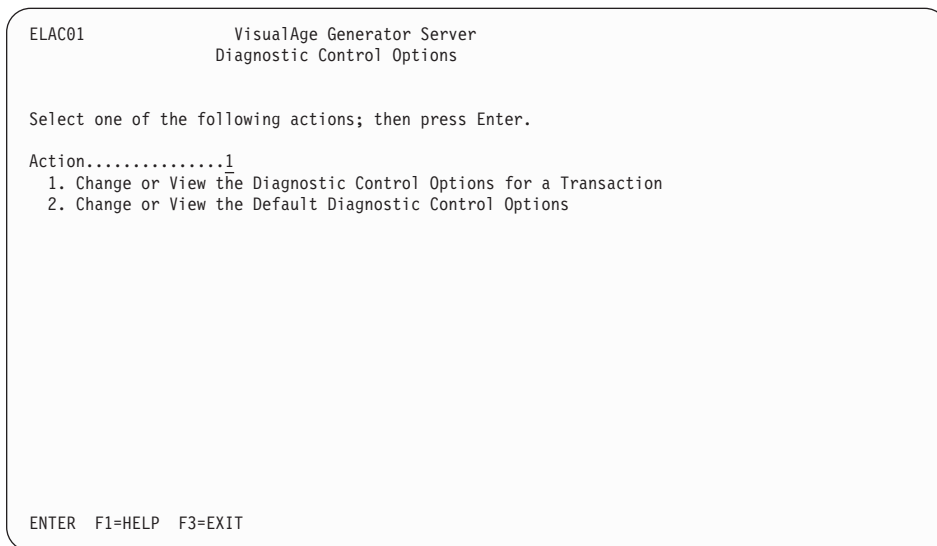


Figure 33. Diagnostic Control Options panel

You can gain access to the following functions from the Diagnostic Control Options panel:

Change or View the Diagnostic Control Options for a Transaction

This option enables you to change or view the diagnostic options for a specific transaction code.

Change or View the Default Diagnostic Control Options

This option enables you to change or view the installation default diagnostic options.

This affects transaction codes that are not specifically identified to the diagnostic controller.

Change or View Diagnostic Control Options for a Transaction

This function enables you to change the VisualAge Generator Server error diagnostic and control options in effect for a specific CICS transaction.

To start the function do the following:

1. Select option 1, Change or View the Diagnostic Control Options for a Transaction, from the Diagnostic Control Options panel (Figure 33).
2. Press Enter. The Change or View Diagnostic Control Options for a Transaction panel (Figure 34 on page 191) is displayed.

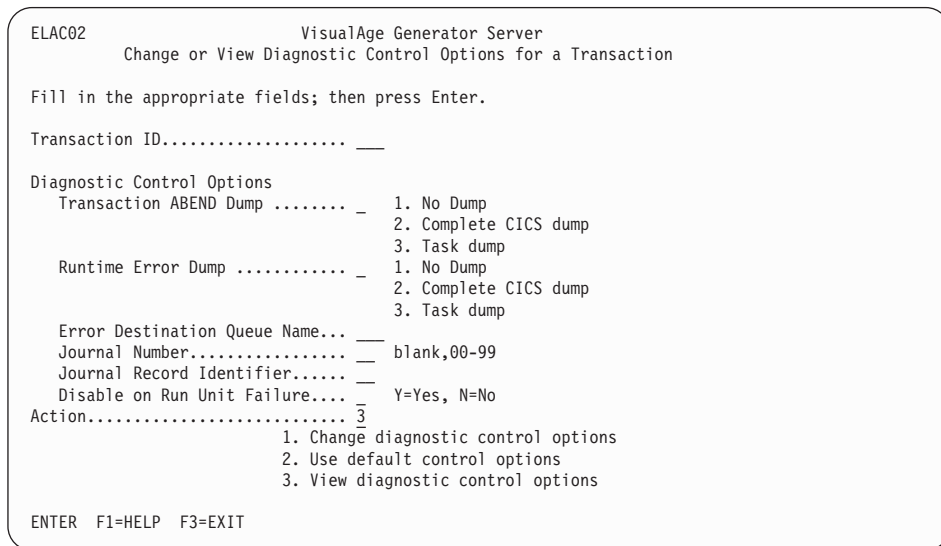


Figure 34. Change or View Diagnostic Control Options for a Transaction panel

The following fields can be entered on the Change or View Diagnostic Control Options for a Transaction panel :

Transaction ID

Specifies the 1 to 4 character identifier of the transaction you want to change the diagnostic options for

Diagnostic Control Options

Transaction ABEND Dump

Specifies the type of dump taken on a CICS transaction ABEND

The types of dumps are:

- 1. No Dump
- 2. Complete CICS dump
- 3. Task dump

Runtime Error Dump

Specifies the type of dump taken on a VisualAge Generator Server-detected error for which a dump is indicated in the error message explanation

The types of dumps are:

- 1. No Dump
- 2. Complete CICS dump

3. Task dump

Error Destination Queue Name

Specifies the 1 to 4 character name of a transient data queue to which VisualAge Generator Server error diagnostic messages are written whenever a transaction ends abnormally due to an error

If this field is blank, no messages are written to a queue.

Journal Number

Specifies the journal number of the CICS journal to which error diagnostic messages are written whenever a transaction is not successful due to an error

If this field is blank, no journal messages are written.

Journal Record Identifier

Specifies the 1 to 2 character record identifier used when messages are written to the CICS journal

If this field is blank, the default identifier EZ is used.

Disable on Run Unit Failure

Specifies whether a transaction is disabled whenever an error is detected that is likely to occur each time the transaction is run

Y Specifies that the transaction is disabled when these errors are detected

N Specifies that the transaction is not be disabled

Action

Allows you to change the current options, view the current options, or accept the default options

To change the options currently set for a transaction do the following:

1. Specify the transaction identifier and any changes.
2. Select 1, Change diagnostic control options.
3. Press Enter.

To use the installation defaults for the transaction do the following:

1. Type the transaction identifier.
2. Select 2, Use default control options.
3. Press Enter.

To view the options currently set for a transaction do the following:

1. Type the transaction identifier.

2. Select 3, View diagnostic control options.
3. Press Enter.

Change or View Default Diagnostic Control Options

This function enables you to change or view the default diagnostic options for transactions that are not identified to the diagnostic controller. If your default options were not modified at installation, the default diagnostic options are set as follows:

- Transaction ABEND and runtime errors both cause a task dump.
- The error destination queue name is ELAD.
- Diagnostic messages are not written to a CICS journal data set.
- Transactions are not disabled on a run unit error.

To start this function do the following:

1. Select 2, Change or View the Default Diagnostic Control Options, from the Diagnostic Control Options panel (Figure 33 on page 190).
2. Press Enter. The Change or View Default Diagnostic Control Options panel is displayed:

```

ELAC04                               VisualAge Generator Server
                                Change or View Default Diagnostic Control Options

Fill in the appropriate fields; then press Enter.

Default Diagnostic Control Options

Transaction ABEND Dump ..... 3   1. No Dump
                                      2. Complete CICS dump
                                      3. Task dump
Runtime Error Dump ..... 3       1. No Dump
                                      2. Complete CICS dump
                                      3. Task dump
Error Destination Queue Name... ELAD
Journal Number..... blank b1ank,00-99
Journal Record Identifier..... EZ
Disable on Run Unit Failure.... N  Y=Yes, N=No

ENTER F1=HELP F3=EXIT

```

Figure 35. Change or View Default Diagnostic Control Options

The options on this panel are the same as those defined for changing or viewing the diagnostic control options for a transaction. They are all defined following Figure 34 on page 191.

Chapter 22. Using the Message File Conversion Utility

┌ **General-Use Programming Interface** _____

The message file conversion utility enables you to create an external source format message file from an existing message file. Once the message file has been converted to external source format, the file can be imported into VisualAge Generator to create the message table with its table contents. You can transfer message tables using the external source format import and export. Refer to the online help system provided with VisualAge Generator Developer for more information about the import and export facility.

└ **End of General-Use Programming Interface** _____

Converting a Message File to an External Source Format Table

┌ **General-Use Programming Interface** _____

Perform the following procedures to convert an existing message file into an external source format table:

1. Allocate file MSGFILE to an existing message file.
2. Allocate file MSGTBLE to a sequential file to receive the external source format. The file must have 80-byte fixed-length records.
3. Include the conversion parameter statement in your job stream. Figure 71 on page 390 shows the syntax of the conversion parameter statement.
4. Start the message utility, DZGMSGC.

└ **End of General-Use Programming Interface** _____

┌ **General-Use Programming Interface** _____

Once the external source format is created using the message file conversion utility, import the file. This will put the message table part into the library. You can then generate the tables to use when you run your programs. Figure 37 on page 198 shows sample JCL that can be used to run the conversion utility.

└ **End of General-Use Programming Interface** _____

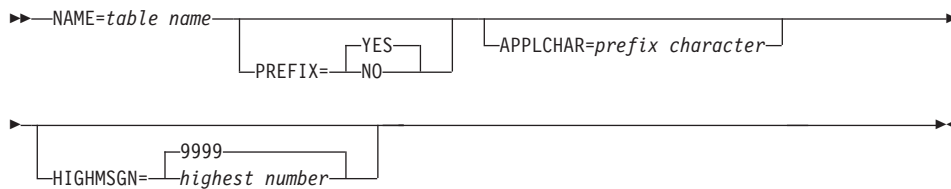


Figure 36. Message Table Conversion Parameters

NAME

Specifies the name of the table part to be created for VisualAge Generator Developer.

This keyword is required. The name can be 6 or 7 characters long, starting with an alphabetic character (A-Z). The remaining characters can be any combination of alphanumeric characters (A-Z, 0-9). The format of the name is `vvvccc`, where `vvvv` is the value of the message table field specified in the program specifications, and `ccc` is a language code.

Valid language codes are as follows:

- ENU** U.S. English
- ENP** Uppercase English
- CHS** Simplified Chinese
- DEU** German
- DES** Swiss German
- ESP** Spanish
- JPN** Japanese (Kanji)
- KOR** Korean
- PTB** Brazilian Portuguese

PREFIX

Specifies whether the prefix should be included in the message text.

- YES** Specifies that the prefix should be included in the message text. The default is YES.
- NO** Specifies that the prefix should not be included in the message text.

APPLCHAR

Specifies the character that is to be inserted before the message number. If a character is not specified, the prefix is created with a blank before the message number. In message files, this character

corresponds to the fourth character in the Message File field of the program specifications, or if there are less than 4 characters in this field, a blank.

HIGHMSGN

Specifies the highest message number to be converted. Valid values are 1 through 9999. The default is 9999. If the highest number is not known or you do not specify this parameter, the utility converts all messages in the message file.

The message conversion utility returns the following codes:

- 0** Message file conversion was successful
- 8** Error occurred during message conversion.

The return code is passed in register 15 through EZERCODE. This code can be checked for conditional execution of later steps in a job.

Sample JCL for the Message File Conversion Utility

In the sample JCL shown in Figure 37 on page 198, the conversion parameter statement is included following the MSGOPTS DD statement that is near the end of the JCL.

```

//DZGMSGC JOB
//*-----
//*
//*   SAMPLE JCL FOR RUNNING THE MESSAGE FILE CONVERSION UTILITY
//*
//*-----
//*
//*-----
//*   ONE OF THE THREE CHARACTER NLS CODES CAN BE SPECIFIED
//*   AS A PARM.  IF NO NLS CODE IS SPECIFIED, THE
//*   INSTALLATION DEFAULT IS USED
//*-----
//ONE EXEC PGM=DZGMSGC,PARM='ENU'
//*
//*-----
//*   LOAD LIBRARIES
//*-----
//STEPLIB DD DSN=ELA.V1R2M0.SELALMD,DISP=SHR
//          DD DSN=CEE.SCEERUN,DISP=SHR
//*
//*-----
//*   PRINT FILES
//*-----
//SYSOUT  DD SYSOUT=A
//SYSOUT  DD SYSOUT=A,DCB=(RECFM=FB,BLKSIZE=133)
//EZPRINT DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=654,BLKSIZE=658)
//SYSPRINT DD SYSOUT=A
//*-----
//*   EXISTING MESSAGE FILE - REPLACE WITH YOUR DATASET NAME
//*-----
//MSGFILE DD DSN=USER1.MSGFILE,DISP=SHR
//*-----
//*   FILE TO RECEIVE DEFINITION IN EXTERNAL SOURCE FORMAT.
//*   IF NOT ALLOCATED, YOU SHOULD CREATE IT FIRST
//*-----
//MSGTBLE DD DSN=USER1.MSGTBLE,DISP=MOD
//*-----
//*   OPTIONS FILE
//*   VALID OPTIONS ARE:
//*   NAME=      REPRESENTS NAME OF THE TABLE PART
//*   PREFIX=    WHETHER OR NOT A MESSAGE PREFIX IS DESIRED
//*   APPLCHAR=  APPLICATION SPECIFIC CHARACTER FOR INSERT
//*   HIGHMSGN=  HIGHEST MESSAGE NUMBER TO CONVERT
//*-----
//MSGOPTS DD *
NAME=MSGENU PREFIX=YES APPLCHAR=A HIGHMSGN=9999
/*
//

```

Figure 37. Sample JCL for Message Conversion Utility

Chapter 23. Using the Parameter Group Utility

Use the parameter group utility to create and maintain the parameter groups in the parameter group file. Each group contains parameters for controlling terminal printer utility (FZETPRT) transactions.

See “Special Parameter Group for the FZETPRT Program” on page 59 for a description of the startup parameters that can be included in the parameter group used with the FZETPRT program.

You can use the parameter group utility to perform the following operations:

- Display the contents of existing parameter groups
- View a list of existing parameter group names
- Add a new parameter group
- Change a parameter group
- Delete a parameter group

Table 27 shows the steps used to define a parameter group file.

Table 27. Defining Parameter Group Files for MVS CICS

Procedure

1. Define the parameter group file using the IDCAMS utility.
DEFINE CLUSTER (NAME(PARM.GROUP.FILE)-
RECORDS(100 100) KEYS(16 0) RECORDSIZE(272 272) INDEXED)
2. Initialize the parameter group file by using the IDCAMS REPRO function to insert a dummy record into the file.
3. Specify the FCT for the parameter group file utility to have access to a user-defined message file for MVS CICS.
DFHFCT TYPE=DATASET, C
DATASET=EZEPRMG, C
ACCMETH=VSAM, C
SERVREQ=(READ,UPDATE,ADD,DELETE,BROWSE), C
FILESTAT=(ENABLED,CLOSED), C
RECFORM=FIXED C
4. Allocate the file by adding the following statement to the MVS CICS startup JCL:
//EZEPRMG DD DISP=SHR,DSN=PARM.GROUP.FILE

Note: The name that designates the parameter group file (EZEPRMG) is a reserved file name and cannot be used as a data file by a VisualAge Generator Developer-defined program.

When the file has been created and allocated, you can gain access the parameter group utility by doing the following:

1. Log on to CICS.
2. Type ELAP on a clear screen.
3. Press Enter.

The parameter group utility does not give message-specific tutorial help after a message is displayed and PF1 is pressed. See “Messages” on page 590 for more information and the user response for parameter group utility messages.

Once the parameter group utility has been started, the Parameter Group Specification panel (Figure 38) is displayed. You can specify the parameter group name on this panel.

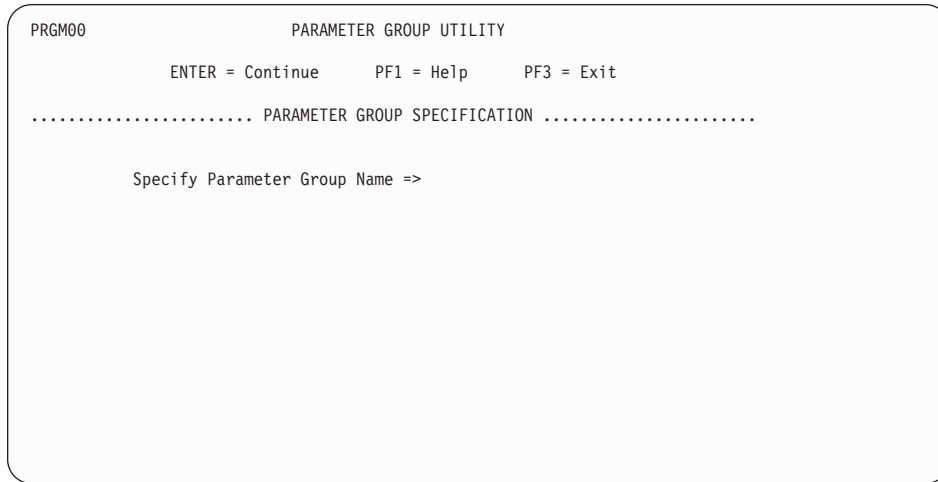


Figure 38. Parameter Group Specification panel

The parameter group name can be from 1 to 4 alphanumeric characters and must be the name of the transaction that was used to start the FZETPRT program. (The utility does not verify this.)

You can enter a group name that already exists if you want to modify a parameter group, or you can enter one that does not exist if you want to define a new parameter group.

Entering a question mark (?) as the group name on the Parameter Group Specification panel displays a list of previously-defined group names on the next panel, the Parameter Group List Display panel (Figure 39). Entering some characters followed by an asterisk (*) displays a list of parameter group names

that begin with the characters that you entered. Entering a specific parameter group name displays the Parameter Group Definition panel (Figure 40 on page 202).

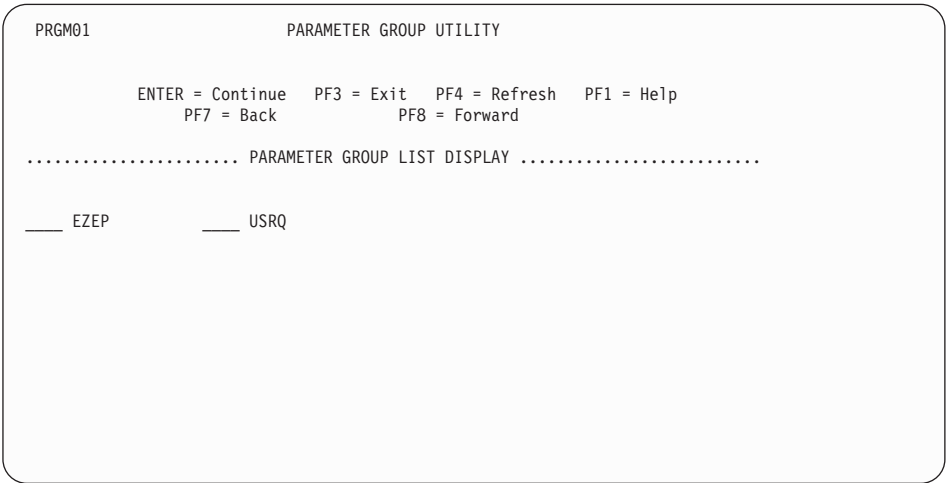


Figure 39. Parameter Group List Display panel

From the Parameter Group List Display panel, you can select a group name to edit by typing an S in the selection field to the left of the group name. You can delete a group by typing a D in the selection field to the left of the group name.

If the specified parameter group already exists, its contents are displayed on the Parameter Group Definition panel. The parameter group can be altered. If the specified parameter group does not exist, the Parameter Group Definition panel is displayed without any data. You can define the new contents; up to 256 characters of data can be entered for a parameter group.

```
PRGM02                PARAMETER GROUP UTILITY

                    PA2 = Cancel  PF1 = Help  PF3 = File and Exit
                    Parameter Group = CCCCCC

.....PARAMETER GROUP DEFINITION.....

Parameter Group:

=>M=PRTBUF=2048 PRTMPP=132 PRTTYP=D FORMFD=NO
```

Figure 40. Parameter Group Definition panel

The parameter group utility does not validate or format the parameters that are specified on the Parameter Group Definition panel. Any parameters that are not valid are ignored when the FZETPRT program is started. See “Special Parameter Group for the FZETPRT Program” on page 59 for more information about setting the parameters for terminal printing.

If you press PF3 on the Parameter Group Definition panel without entering any parameters, a parameter group is stored without any associated parameters. You can store an empty parameter group to reserve parameter group names.

Empty parameter groups do not affect the initialization of the FZETPRT program.

The parameter group utility left-justifies the parameter group name and pads it to the right with blanks (X'40'). The parameter group utility uses this name as a key to index the parameter group file.

If you selected a parameter group from the Parameter Group List Display panel (Figure 39 on page 201), after the Parameter Group Definition panel is processed, the Parameter Group List Display panel is displayed again with the original request replaced by an asterisk beside the group name that was processed. An asterisk (*) is ignored as input on the Parameter Group Definition panel if more processing is done.

Chapter 24. IMS Diagnostic Message Print Utility

When a generated program ends abnormally due to an error condition in IMS environments, diagnostic error messages are written to the message queue identified by the /ERRDEST generation option specified for the first program in the run unit.

A BMP is provided to print the messages in the message queue. The JCL needed to print the diagnostic information is supplied as member ELAMQJUD of ELA.V1R2M0.ELAJCL (see Figure 41).

The msg-queue identified by the IN parameter is the name of the queue that was specified for ERRDEST when the program was generated.

```

//*****00000100
//** ELAMQJUD - JCL TO DRAIN AND PRINT THE ELADIAG MESSAGE QUEUE 00000200
//**          FOR VISUALAGE GENERATOR SERVER. 00000300
//**          THIS PROGRAM RUNS AS A BMP. 00000400
//** 00000500
//** LICENSED MATERIALS - PROPERTY OF IBM 00000600
//** 5648-B02 (C) COPYRIGHT IBM CORP. 1994, 1998 00000700
//** SEE COPYRIGHT INSTRUCTIONS 00000800
//** 00000900
//** STATUS = VERSION 1, RELEASE 2, LEVEL 0 00001000
//** 00001100
//** TO TAILOR THIS JOBSTREAM: 00001200
//**     1. COPY A JOBCARD. 00001300
//**     2. CHANGE IN= TO THE NAME OF YOUR ERROR DIAGNOSTIC 00001400
//**        QUEUE. 00001500
//**     3. MAKE SURE THAT THE TRANSACTION SPECIFIED BY IN= 00001600
//**        AND THE ELAMPUTL PROGRAM ARE STARTED BY IMS. 00001700
//** 00001800
//** RETURN CODES 00001900
//**     0 - SUCCESSFUL COMPLETION 00002000
//**     4 - NO MESSAGES ON QUEUE TO DRAIN. 00002100
//**     16 - FATAL ERROR. PROCESSING TERMINATED 00002200
//**     20 - OPEN FAILED ON ELAPRINT 00002300
//** 00002400
//*****00002500
//DRAINMQ EXEC IMSBATCH,MBR=ELAEPUTL, 00002600
//          PSB=ELAMPUTL,IN=ELADIAG,RGN=4096K 00002700
//G.STEPLIB DD 00002800
//          DD 00002900
//          DD DSN=CEE.SCEERUN,DISP=SHR 00003000
//          DD DSN=ELA.V1R2M0.SELALMD,DISP=SHR 00003100
//G.ELAPRINT DD SYSOUT=* 00003200
//G.SYSOUT DD SYSOUT=* 00003300
//G.SYSPRINT DD SYSOUT=* 00003400
/* 00003500

```

Figure 41. ELAMQJUD

Part 5. Diagnosing Problems on MVS Systems

Chapter 25. Diagnosing Problems for VisualAge Generator Server on MVS Systems

- Detecting Errors 207
 - Screen Variable Edit Errors 207
 - File and Database Errors—Category 1 208
 - File and Database Errors—Category 2 208
 - File and Database Errors—Category 3 209
 - Arithmetic Overflow Errors 209
 - Logic Errors 210
 - Program Interaction Errors 211
 - System Errors 213
- Reporting Errors 213
 - Controlling Error Reporting in MVS CICS 214
 - Controlling Error Reporting in IMS Environments 214
 - Controlling Error Reporting in MVS/TSO and MVS Batch Environments 215
 - Error Reporting Summary 215
 - Transaction Error 215
 - Rununit Error 216
 - Catastrophic error 218
 - VisualAge Generator Server Error 219
 - Using the VisualAge Generator Server Error Panel 220
 - Printing Diagnostic Information for IMS
 - ERRDEST Message Queue 221
 - IMS Log Format 222
 - Running the Diagnostic Print Utility 224
 - Printing Diagnostic Information for MVS CICS 224
 - MVS CICS Diagnostic Message Layout 224
 - Running the Diagnostic Print Utility 226
 - Analyzing Errors Detected while Running a Program 226
- Finding Information in Dumps 227
 - VisualAge Generator Server ABEND Dumps 227
 - COBOL or Subsystem ABEND Dumps 228
 - Information in the VisualAge Generator Server Control Block 229
 - Information in an Application, Print Services, or Table Profile Block 229

- How to Find the Current Position in a Program at Time of Error 230
- VisualAge Generator Server Trace Facility 230
 - Enabling VisualAge Generator Program Source-Level Tracing with a Generation Option 231
 - Activating a Trace 232
 - Activating a Trace Session for MVS CICS or IMS/VS 232
 - Activating a Trace Session for MVS/TSO, MVS Batch, or IMS BMP 236
 - Deactivating a Trace Session 238
 - Printing Trace Output 239
 - Printing the Trace Output in MVS CICS 239
 - Printing the Trace Output in MVS/TSO, MVS Batch, and IMS BMP 239
 - Printing the Trace Output in IMS/VS 239
- Reporting Problems for VisualAge Generator Server 240
- Problem Reporting Sequence 242

Chapter 26. Common Messages during Preparation for MVS Systems

- Common Abend Codes during Preparation 245
- MFS Generation Messages 245
- DB2 Precompiler and Bind Messages 246
- DB2/VSE Precompiler Messages 247
- COBOL Compilation Messages 248

Chapter 27. Common System Return Codes for MVS Systems

- Common EZERT8 Return Codes 251
 - System Return Code Formats for EZERT8 252
 - Common System Return Codes in EZERT8 254
 - VisualAge Generator Server Return Codes 256
- Common SQL Return Codes 266
- Common DL/I Status Codes 268
- Common VSAM Status Codes 269
 - OPEN request type 269
 - CLOSE request type 270

GET/PUT/POINT/ERASE/CHECK/ENDREQ request types	270
COBOL Status Key Values.	271

**Chapter 28. VisualAge Generator Server
Return Codes and Abend Codes for MVS**

Systems	273
Return Codes	273
ABEND Codes	273
MVS CICS Environments	274
IMS, IMS BMP, MVS/TSO, and MVS Batch Environments.	276

**Chapter 29. Codes from Other Products
for MVS Systems**

Common System Abend Codes for All Environments	279
LE Runtime Messages	281
COBOL Runtime Messages	282
Common COBOL Abend Codes.	282
Common IMS Runtime Messages	283
Common IMS Runtime Abend Codes	284
Common MVS CICS Runtime Messages	285
Common MVS CICS Abend Codes.	286
COBOL Abends under CICS	287

Chapter 25. Diagnosing Problems for VisualAge Generator Server on MVS Systems

The chapter contains diagnosis, modification, or tuning information. Use this information to determine the source of the problem you encountered. Some common program definition, database, and system errors that might cause problems are described. This chapter also explains how to obtain error listings and diagnose runtime errors.

Detecting Errors

You can find most logic errors by using the test facility before you generate your program.

During generation, the preprocessor checks your program for any remaining syntax errors. In addition, VisualAge Generator also checks that your use of language elements is consistent with the resource association information you select for each file. For example, you can SCAN a serial file in the IMS/VS environment only if it is associated with PCB number 0 and if the program is a batch program.

The VisualAge Generator Developer detects a few additional errors. For example, an error occurs if there is not enough space on a map for the IMS transaction code or the flag that indicates the type of information that is stored in the work database.

When you run your generated program, different types of errors are detected by VisualAge Generator Server, COBOL, the subsystem (IMS or MVS CICS), or MVS. The error handling varies depending on which product detects the error. Table 28 on page 208 shows the types of errors that can occur when you run your generated program, as well as VisualAge Generator Server functions you can use for error detection and reporting.

Screen Variable Edit Errors

The end user entered data that is not valid. For example:

- Alphabetic data in numeric field
- Value not in allowable range
- Omitted a required field

Table 28. Types of Errors

VisualAge Generator Set Controls	Error Detected By	Error Handling
Use map definition to specify variable field edits or to include EZEMSG on the map.	Program logic or VisualAge Generator Server.	The error message is displayed on the map (in EZEMSG) and the cursor is placed on the field in error. If EZEMSG is not provided on the map, the ELAM01 screen is displayed with the EZEMSG error message.

File and Database Errors—Category 1

A hard error occurred in a file or database I/O operation. For example:

- DL/I call not compatible with the processing options in the IMS PSB
- I/O error on the file or database
- File not found

Table 29. Types of Errors

VisualAge Generator Set Controls	Error Detected By	Error Handling
Use EZEFEFC to indicate whether the program provides logic to override normal database and file error processing for hard errors. Hard errors are defined in the online helps within VisualAge Generator Developer	Access method or database manager. The access method or database manager return an error code to the generated logic or VisualAge Generator Server.	If EZEFEFC is set to 0 and a hard error occurs, the program ends with a transaction error, as described in Table 36 on page 216 . For SQL errors, the DB2 messages describing the SQL error code are obtained from DB2. Refer to the DB2 messages manual for your system for explanations of the error codes. If EZEFEFC is set to 1 and a hard error occurs, the program is responsible for controlling the error handling.

File and Database Errors—Category 2

A file or database error occurs due to incorrect program logic. For example, deleting or replacing a record without first doing an UPDATE process option

is an example of this error type.

Table 30. Types of Errors

VisualAge Generator Set Controls	Error Detected By	Error Handling
Follow the file management rules in the VisualAge Generator help facility.	VisualAge Generator Server.	The program ends with a run unit error, as described in Table 36 on page 216.

File and Database Errors—Category 3

Two generated programs which are being used together are defined in an inconsistent manner. For example, two programs have been generated with different file definitions.

Table 31. Types of Errors

VisualAge Generator Set Controls	Error Detected By	Error Handling
Follow the rules in the VisualAge Generator help facility.	COBOL or VisualAge Generator Server.	If detected by VisualAge Generator Server, the program ends with a run unit error, as described in Table 36 on page 216. If detected by COBOL and either STAE is specified or you are using LE, a formatted dump is generated. If detected by COBOL and NOSTAE is specified, a system dump is generated.

Arithmetic Overflow Errors

One of three situations occurs:

- Maximum value overflow when the result of an arithmetic operation does not fit into the maximum size number (18 digits)
- User variable overflow when the result of an arithmetic operation or a move to a numeric data item cause significant value (not decimal positions) to be lost
- Division by 0

Table 32. Types of Errors

VisualAge Generator Set Controls	Error Detected By	Error Handling
<p>Use the /NUMOVFL generation option to indicate that the generated program should check for overflow.</p> <p>If /NUMOVFL is specified, use the EZEOPER special function word to indicate whether program processing should continue if an overflow occurs.</p>	<p>If /NUMOVFL is specified, special logic is generated to intercept numeric overflow conditions. The EZEOPER special function word can be used to indicate the program provides logic to handle the overflow.</p> <p>If /NONUMOVFL is specified, overflow errors are not detected. The contents of the EZEOPER special function word are ignored. Truncation occurs on maximum value overflow, or user variable overflow. An abend occurs on division by 0.</p>	<p>If /NUMOVFL is specified and the EZEOPER special function word is set to 0 or 1, the maximum value overflow causes the program to end abnormally with a transaction error, as described in Table 36 on page 216 .</p> <p>If /NUMOVFL is specified and the EZEOPER special function word is set to 1, the user variable overflow also causes the program to end abnormally with a transaction error.</p> <p>In other situations, if /NUMOVFL is specified, the program continues to run.</p> <p>If /NONUMOVFL is specified, overflow errors are not detected. Division by 0 causes an abend (0C7 for non-MVS CICS systems and ASRA for MVS CICS systems). You need to code the program to ensure that overflow does not occur.</p>

Logic Errors

The program ends because of faulty logic. For example:

- Numeric data that is not initialized
- Problems with array subscripting
- Infinite loops
- Moving incompatible data

Table 33. Types of Errors

VisualAge Generator Set Controls	Error Detected By	Error Handling
Specifying the /INITRECD and /INITADDWS options force numeric data in records to be initialized during program initialization. However, it is still possible that numeric data in files or databases contains non-numeric values.	<p>The test facility can be used to detect most of these errors before generation.</p> <p>If the COBOL STAE runtime option or the LE TRAP(ON) runtime option is specified, COBOL traps most of these errors.</p> <p>Otherwise, MVS CICS, IMS, or the MVS operating system detects the errors.</p>	<p>In MVS CICS, the EXEC CICS HANDLE ABEND function gives control to the generated program error handler. The error is then treated as a run unit error as described in Table 36 on page 216 .</p> <p>In other systems:</p> <ul style="list-style-type: none"> • If STAE is specified, or you are using LE, a formatted dump is generated. • If NOSTAE is specified, a system dump is generated.

Program Interaction Errors

Two generated programs that are used together are defined or run in an inconsistent manner; or, all parts of a generated program were not installed correctly.

For example:

- A called program has been passed too few parameters.
- A program with a First Map specification is being transferred to using a DXFR statement.
- A transferred to program receives a map but does not have the First Map specification.
- The program linkage table did not specify the same linkage conventions when the calling and called programs were generated.

MVS CICS specific examples include the following:

- A program does a CONVERSE or DISPLAY process with a map, but the online print services program or the map group format module were not defined in the MVS CICS PPT tables.
- A program attempts to transfer to another program with a dynamic CALL or a DXFR statement, but the transferred to program was not defined in the MVS CICS PPT tables.
- A program attempts to gain access to a VisualAge Generator table, but the table program was not defined in the MVS CICS PPT tables.

- A program attempts to transfer to another program with an XFER statement, but the transferred-to transaction code was not defined in the MVS CICS PCT tables.

IMS specific examples include the following:

- A program receives a message segment that is not valid.
- A program receives a sequence of messages from IMS that is not valid.
- A program attempts to send messages that IMS rejects.
- A program attempts to transfer to another program with an XFER statement, but the transferred-to transaction code was not defined in the IMS system generation.

Note: If an XFER statement with a map is used for a nonconversational IMS/VS program, IMS detects this error.

Table 34. Types of Errors

VisualAge Generator Set Controls	Error Detected By	Error Handling
Follow the rules for the XFER and DXFR statements in the VisualAge Generator help facility. For MVS CICS, follow the guidelines in Chapter 15. Preparing and Running a Generated Program in MVS CICS. For MVS/TSO, follow the guidelines in Chapter 16. Preparing and Running a Generated Program in MVS/TSO. For MVS batch, follow the guidelines in Chapter 17. Preparing and Running Generated Programs in MVS Batch. For IMS, follow the guidelines in Chapter 18. Preparing and Running Generated Programs in IMS/VS and IMS BMP. If special linkage conventions are used, ensure that the same linkage table is used for generating the calling and called programs.	VisualAge Generator Server.	The program ends with a run unit error, as described in Table 36 on page 216.

System Errors

The program was not properly installed. Common mistakes are the following:

- Missing PSB or ACB
- JCL errors in batch programs
- MVS/TSO CLIST errors
- Errors in MVS CICS table definition
- Errors in IMS system definition
- Module cannot be found
- Storage cannot be obtained

Table 35. Types of Errors

VisualAge Generator Set Controls	Error Detected By	Error Handling
For MVS CICS, follow the guidelines in “Chapter 15. Preparing and Running a Generated Program in MVS CICS” on page 143. For MVS/TSO, follow the guidelines in “Chapter 16. Preparing and Running a Generated Program in MVS/TSO” on page 151. For MVS batch, follow the guidelines in “Chapter 17. Preparing and Running Generated Programs in MVS Batch” on page 159. For IMS, follow the guidelines in “Chapter 18. Preparing and Running Generated Programs in IMS/VS and IMS BMP” on page 165 .	IMS, MVS CICS or MVS operating system.	The program ends abnormally with an MVS CICS dump in MVS CICS systems or a system abend in IMS/VS, IMS BMP, MVS/TSO, or MVS batch systems. For MVS CICS, dump production is controlled using the online diagnostic controller utility (“Chapter 21. Diagnostic Control Options” on page 189); for non-MVS CICS systems, a system abend dump is produced if there are SYSABEND or SYSUDUMP statements in the JCL or CLIST.

Reporting Errors

VisualAge Generator Server provides functions that help you determine the cause of a runtime problem. All runtime errors that VisualAge Generator Server traps are accompanied by error messages and supporting information to help diagnose the problem. Table 36 on page 216 shows the error diagnostic actions that can be taken based on the severity of the error and type of system being used.

Controlling Error Reporting in MVS CICS

In the MVS CICS environment, error actions are controlled through the online diagnostic controller utility installed as transaction ELAC.

The utility allows you to specify what type of dump is requested, the name of the transient data queue to which VisualAge Generator Server diagnostic messages are written, the CICS journal number and identifier for error messages, and whether or not a transaction is disabled when a run unit error is detected. The utility lets you reset the default options for all transactions and override the default options for individual transactions.

See “Chapter 21. Diagnostic Control Options” on page 189 for more details about the diagnostic controller utility.

Controlling Error Reporting in IMS Environments

The following error actions are controlled by generation options for the IMS/VS and IMS BMP environments:

- Write error messages to the error destination. The destination is determined by the /ERRDEST generation option.
- Write error messages to the system log. The log ID is determined by the /LOG generation option. /NOLOG indicates that error messages should not be written to the system log.
- Put the message that caused the problem for transaction-oriented IMS BMP programs back on the message queue. /RECOVERY indicates that the message being processed when the error occurred should be placed back on the message queue before the program ends. /NORECOVERY indicates that the message being processed should be deleted and not placed back on the message queue. This option is applicable only to a run unit error when VisualAge Generator Server detects the error. It does not apply to transaction-oriented BMPs that use CSPTDLI to read the message queue.
- Issue ROLL call or abend for a run unit error. /NOFASTPATH results in a ROLL call. /FASTPATH results in a 1602 abend.

The actions controlled by the runtime JCL are as follows:

- Print message. This is done only if there is an ELAPRINT DD statement in the runtime JCL.
- Snap dump. If the message indicates a snap dump is taken, the snap dump is produced only if there is an ELASNAP DD statement in the runtime JCL.
- Abend 1602 or 1600. This creates a dump only if the runtime JCL contains a SYSUDUMP or SYSABEND DD statement.

Abend code 1602 is the user code issued by VisualAge Generator Server when it ends the run unit for a /FASTPATH program because of an error.

Abend code 1600 is the user code issued by VisualAge Generator Server in all other situations when it ends program processing because of an unrecoverable error.

IMS takes the following actions, based on the way VisualAge Generator Server ends the program:

- If a rollback (ROLB) call is issued, the database changes are backed out, the logical unit of work ends, the next message is read from the message queue, and processing continues.
- If a ROLL call is issued, the database changes are backed out, the logical unit of work ends, and IMS stops the program with a user 778 abend. The transaction and PSB are not stopped and can be scheduled again without operator intervention.
- If either a 1600 or a 1602 abend is issued, the database changes are backed out, the logical unit of work ends, and IMS stops the program. The transaction and PSB are also stopped, and they require operator intervention to start them again.

Use ELASNAP so that sufficient data is captured the first time an error occurs.

Controlling Error Reporting in MVS/TSO and MVS Batch Environments

The actions controlled by the runtime CLIST or JCL are as follows:

- Print message. This is done only if there is an ELAPRINT ALLOCATE command in the runtime CLIST or an ELAPRINT DD statement in the runtime JCL.
- Snap dump. If the message indicates a snap dump is taken, the snap dump is produced only if there is an ELASNAP ALLOCATE command in the runtime CLIST or an ELASNAP DD statement in the runtime JCL.
- Abend 1600. This creates a dump only if the runtime CLIST or JCL contains a SYSUDUMP or SYSABEND DD statement.

Error Reporting Summary

The following tables summarize the error processing actions for VisualAge Generator Server.

Transaction Error

This error affects only the current IMS transaction or MVS CICS task. In IMS/VS, processing continues with the next message. In MVS CICS, the transaction is still available to other end users.

Table 36. Error Processing Actions For VisualAge Generator Server Detected Errors

Environment	Action
MVS/TSO	See run unit error.
MVS CICS	<ul style="list-style-type: none"> • Write error messages to error destination (diagnostic controller option) • Write error messages to MVS CICS journal data set (diagnostic controller option) • MVS CICS dump, dump code ELAD, as determined by message. The type of dump issued for a particular transaction is a diagnostic control option. • Issue a rollback request • Display error messages on terminal, if possible • Set return code to 693 (called programs only)
MVS Batch	See run unit error.
IMS BMP	See run unit error.
IMS/VS (Initial generated program is a main batch program or a called batch program)	See run unit error.
IMS/VS (Initial generated program is a main transaction).	<ul style="list-style-type: none"> • Write error messages to error destination (/ERRDEST generation option) • Write error messages to system log (/LOG generation option) • Print message (ELAPRINT DD statement) • Snap dump determined by the message (ELASNAP DD statement) • Display error messages on current LTERM • Issue a rollback (ROLB) request • Read next message from the queue

Rununit Error

The error is likely to occur for every user. In IMS, a new copy of the program is used if there are additional messages on the queue. In MVS CICS, the transaction might be disabled.

Table 37. Error Processing Actions For VisualAge Generator Server Detected Errors

Environment	Action
MVS/TSO	<ul style="list-style-type: none"> • Print message (ELAPRINT DD statement) • Snap dump determined by the message (ELASNAP DD statement) • Display error messages on terminal, if possible • Issue a rollback request if DL/I or DB2 databases were used • Set return code to 693 • Return
MVS CICS	<ul style="list-style-type: none"> • Write error messages to error destination (diagnostic control option), if possible • Write error messages to MVS CICS journal data set (diagnostic control option), if possible • Disable transaction (diagnostic control option) • MVS CICS dump, dump code ELAD, as determined by message. The type of dump issued for a particular transaction is a diagnostic control option. • Issue a rollback request • Display error messages on terminal, if possible • Set return code to 693 (called programs only) • Return
MVS BATCH	<ul style="list-style-type: none"> • Print message (ELAPRINT DD statement) • Snap dump determined by the message (ELASNAP DD statement) • Issue a rollback request if DL/I or DB2 databases were used • Set return code to 693 • Return
IMS BMP	<ul style="list-style-type: none"> • Write error messages to error destination (/ERRDEST generation option) • Write error messages to system log (/LOG generation option) • Print message (ELAPRINT DD statement) • Snap dump determined by the message (ELASNAP DD statement) • Issue a rollback (ROLB) request • Insert message segment or segments into the queue again (/RECOVERY generation option) • Set return code to 693 • Return

Table 37. Error Processing Actions For VisualAge Generator Server Detected Errors (continued)

Environment	Action
IMS/VS (Initial generated program is a main batch program or a called batch program)	<ul style="list-style-type: none"> • Write error messages to error destination (/ERRDEST generation option), if possible • Write error messages to system log (/LOG generation option), if possible • Print messages (ELAPRINT DD statement), if possible • Snap dump determined by the message (ELASNAP DD statement) • Issue ROLL request is generated with /NOFASTPATH • Abend 1602 if generated with /FASTPATH
IMS/VS (Initial generated program is a main transaction)	<ul style="list-style-type: none"> • Write error messages to error destination (/ERRDEST generation option), if possible • Write error messages to system log (/LOG generation option), if possible • Print message (ELAPRINT DD statement) • Snap dump determined by the message (ELASNAP DD statement) • Display error messages on current LTERM • Issue ROLL request is generated with /NOFASTPATH • Abend 1602 if generated with /FASTPATH

Catastrophic error

This error indicates storage is corrupted or standard error reporting processing ends abnormally.

Table 38. Error Processing Actions For VisualAge Generator Server Detected Errors

Environment	Action
MVS/TSO	<ul style="list-style-type: none"> • Print message (ELAPRINT DD statement), if possible • Display error messages on terminal, if possible • Abend 1600 (SYSUDUMP or SYSABEND DD statement)
MVS CICS	<ul style="list-style-type: none"> • Write error messages to error destination (diagnostic control option), if possible • Write error messages to MVS CICS journal data set (diagnostic control option), if possible • Disable transaction (diagnostic control option) • Display error messages on terminal, if possible • ABEND ELAE. The type of dump issued for a particular transaction is a diagnostic control option.

Table 38. Error Processing Actions For VisualAge Generator Server Detected Errors (continued)

Environment	Action
MVS BATCH	<ul style="list-style-type: none"> • Print message (ELAPRINT DD statement), if possible • Abend 1600 (SYSUDUMP or SYSABEND, DD statement)
IMS BMP	<ul style="list-style-type: none"> • Write error messages to error destination (/ERRDEST generation option), if possible • Write error messages to system log (/LOG generation option), if possible • Print message (ELAPRINT DD statement), if possible • Abend 1600 (SYSUDUMP or SYSABEND DD statement)
IMS/VS (Initial generated program is a main batch program or a called batch program)	<ul style="list-style-type: none"> • Write error messages to error destination (/ERRDEST generation option), if possible • Write error messages to system log (/LOG generation option), if possible • Print messages (ELAPRINT DD statement), if possible • Abend 1600 (SYSUDUMP or SYSABEND DD statement)
IMS/VS (Initial generated program is a main transaction)	<ul style="list-style-type: none"> • Write error messages to error destination (/ERRDEST generation option), if possible • Write error messages to system log (/LOG generation option), if possible • Print message (ELAPRINT DD statement), if possible • Display error messages on current LTERM, if possible • Abend 1600 (SYSUDUMP or SYSABEND DD statement)

VisualAge Generator Server Error

A VisualAge Generator Server error at a point where standard error reporting process is not active.

Table 39. Error Processing Actions For VisualAge Generator Server Detected Errors

Environment	Action
All environments	<ul style="list-style-type: none"> • Abend, ABEND code indicates the reason for the error

See Table 43 on page 228 for information concerning the contents of the registers when either a 1600, 1602, or an ELAE abend occurs.

When your IMS BMP program ends with a run unit or catastrophic error, all updates since the last checkpoint are rolled back and the program ends. You

should include checkpoint and restart logic in the program if it is to run as a BMP. Refer to IMS documentation for your system for additional information about checkpoint and restart.

Using the VisualAge Generator Server Error Panel

When an error occurs, VisualAge Generator Server attempts to display error messages on the current terminal. The panels used in displaying error messages are defined as map group ELAxxx where xxx is the language code. External source format for the map group can be found in ELA.V1R2M0.SELAESF of member ELAxxx, so that you can customize the constant information on your map to your own installation. For example, you can add the phone number of your system administrator so that the user encountering the error knows who to call to report the problem.

Refer to the *Program Directory for VisualAge Generator Server for MVS* for information about tailoring the panel.

The following figure shows the error panel (map ELAM02 in the map group) as it is shipped with the product. The panel shows the same diagnostic information that is written to the error destination queue, system log or journal, or ELAPRINT file. If there are more error messages than can fit on a single panel, the last line on the panel prompts the user to press a key to display additional error messages.

```
Unexpected Program Failure

An unexpected input/output or program error occurred in the
program you were running. Please make a note of the program
name, date, time, and initial error messages and report them to your
system administrator.

Program name ... ART22
Date ..... 08/21/90
Time ..... 13:04:23

Error Messages:

ELA00093I Error occurred in application ART22, process or group ART229
ELA00131P MSGQ error, file = UNKNOWN, function = CHG, status code = A1
ELA00066I DL/I I/O area = UNKNOWN
                        EDDDED4
                        45256650
```

Figure 42. Panel ELAM02 (example).

Printing Diagnostic Information for IMS

Diagnostic messages are sent either to a print file for batch jobs or to a message queue for BMPs or online transactions. A diagnostic utility is provided to print messages written to a message queue. Optionally, based on the /LOG generation option, the diagnostic information can be written to the IMS log.

ERRDEST Message Queue

Table 40 shows the format of the information in the IMS message queue when ERRDEST is used.

Table 40. ERRDEST IMS Message Queue

Field	Length in Bytes	Type of Data	Description
Record length	2	Binary	The length of the record.
Reserved	2	Binary	A reserved field that must contain binary zeros.
IMS transaction code	8	Character	The name used to identify the IMS message queue that was specified with the /ERRDEST generation option.
Date	8	Character	Date of the transaction from the I/O PCB (MM/DD/YY).
Time	8	Character	Time of the transaction from the I/O PCB (HH:MM:SS).
NLS	3		The value specified for the /TARGNLS generation option

Table 40. ERRDEST IMS Message Queue (continued)

Field	Length in Bytes	Type of Data	Description
Message number	9		The message number: Bytes 1-3 Message File Identifier (ELA) Byte 4 Application Identifier (0) Bytes 5-8 Message Number Byte 9 Message Type. A message type of 'C' indicates this record is a continuation of the specified message from a previous record in the queue.
Message number separator (reserved position)			Byte 10 Blank
Message Text	Variable	Character	The text from the message file with specified message inserts.

IMS Log Format

Table 41 shows the format of the information in the IMS log.

Table 41. IMS Log Record

Field	Length in Bytes	Type of Data	Description
Record length	2	Binary	The length of the record.
Reserved	2	Binary	A reserved field that must contain binary zeros.
Log ID	1	Character	The value specified with the /LOG generation option.

Table 41. IMS Log Record (continued)

Field	Length in Bytes	Type of Data	Description
Date	8	Character	Date of the transaction from the I/O PCB (MM/DD/YY).
Time	8	Character	Time of the transaction from the I/O PCB (HH:MM:SS).
NLS	3		The value specified for the /TARGNLS generation option
Message number	9		The message number: Bytes 1-3 Message File Identifier (ELA) Byte 4 Application Identifier (0) Bytes 5-8 Message Number Byte 9 Message Type. A message type of 'C' indicates this record is a continuation of the specified message from a previous record in the queue.
Message number separator (reserved position)			Byte 10 Blank
Message Text	Variable	Character	The text from the message file with specified message inserts.

Running the Diagnostic Print Utility

A BMP is provided to print diagnostic information that is written to the message queue specified by the /ERRDEST generation option. The JCL needed to print the diagnostic information is supplied as member ELAMQJUD of 'ELA.V1R2M0.ELAJCL.

The msg-queue identified by the IN parameter is the name of the queue that was specified in the /ERRDEST option when the application was generated. See "Diagnostic Message Printing Utility" on page 186 for more information.

Printing Diagnostic Information for MVS CICS

Diagnostic messages are sent to a transient data queue for MVS CICS transactions. A diagnostic print utility is provided to print messages written to a transient data queue. Optionally, as specified by the diagnostic controller utility, the diagnostic information can also be written to an MVS CICS journal data set.

MVS CICS Diagnostic Message Layout

Table 42 shows the format of the information in each error message record written to a transient data queue or MVS CICS journal.

Table 42. Diagnostic Message Layout

Field	Length in Bytes	Type of Data	Description
SYSID name	4	Character	The name of the MVS CICS system that the error message was created on.
TRANID name	4	Character	The name of the MVS CICS transaction code that started the logical unit-of-work.
Task identifier	8	Character	The task identifier assigned by MVS CICS to each transaction instance that is processed. This number is reset to 0 when MVS CICS is cold-started. This is taken from EIB field EIBTASKN.

Table 42. Diagnostic Message Layout (continued)

Field	Length in Bytes	Type of Data	Description
ERRDEST name	4	Character	The name of the MVS CICS transient data queue. This field is blank if the record is written to the MVS CICS journal.
Date	8	Character	Date of the transaction (MM/DD/YY)
Time	8	Character	Time of the transaction (HH:MM:SS)
NLS	3	Character	The value specified for the /TARGNLS generation option
Message number	9	Character	The message number: Bytes 1-3 Message File Identifier (ELA) Byte 4 Application Identifier (0) Bytes 5-8 Message Number Byte 9 Message Type. A message type of 'C' indicates this record is a continuation of the specified message from a previous record in the queue.
Message number separator (reserved position)			Byte 10 Blank
Message text	110	Character	The text from the message file with specified message inserts

Running the Diagnostic Print Utility

Use the ELAU transaction to print the messages routed to a transient data queue. You enter the following information on the ELAU panel:

Error destination queue name

The name of the queue containing the messages. The default name is ELAD.

JES spool file information

The node, user ID, and class to which the messages on the queue are spooled. The default class is A and the default node and user ID are * which routes the printed messages to the local spool printer.

Clear queue

Y specifies the queue is deleted after its contents are printed. N specifies the messages are left in the queue after they are printed. Y is the default value.

See “Diagnostic Message Printing Utility” on page 186 for more information about running the diagnostic print utility.

Analyzing Errors Detected while Running a Program

Use the error messages, diagnostic messages, and the information in Chapter 61. VisualAge Generator Server Messages for MVS, VSE, and VM Systems to determine the cause of the problem. If the error is detected by another product (for example, COBOL or IMS), check the information in Chapter 27. Common System Return Codes for MVS Systems and Chapter 29. Codes from Other Products for MVS Systems and the documentation for the other product. If you cannot determine the cause of the problem using this information and if the problem can be created again in the test environment, use the test facility to isolate and debug the error before generating your program.

For environmental debugging, you can use batch terminal simulator (BTS) II for IMS programs or the runtime diagnostic facility (EDF) for MVS CICS programs. In addition, if you use the TEST COBOL compile option, you can use the COBOL debugging facilities.

For more information about testing with the VisualAge Generator test facility, refer to the VisualAge Generator help facility. Refer to the BTS II, MVS CICS, and COBOL manuals for your versions of these products for additional information on their debugging facilities.

If you get a JCL error for the runtime JCL, check the generation listing for the programs involved for any error messages related to JCL generation. In

addition, ensure the tailoring that was done on the JCL templates and VisualAge Generator Server procedures is correct. Also check the changes you made when you customized the sample JCL.

If you get a CLIST error for the runtime CLIST, follow the same process as for a JCL error.

When abends occur, the problem determination might require assistance from the IBM Support Center. In this case, be prepared to provide IBM with the following information:

- VisualAge Generator Developer print of the problem program
- COBOL compilation listing created using the /COMMENTLEVEL=STATEMENTS generation option
- Formatted dump
- VisualAge Generator Server diagnostic information written to either the error diagnostic queue or listed in the printout for ELAPRINT
- IMS log or MVS CICS journal, as appropriate

IBM requests a COBOL debugger trace listing only if the information is needed for problem determination. IBM will give you the information on how to specify the trace options if the information is necessary.

Finding Information in Dumps

Information about the problem program can be determined by finding the address of the VisualAge Generator Server control block in a dump.

VisualAge Generator Server ABEND Dumps

If the dump code is 1600, 1602, or ELAE, the dump was initiated because VisualAge Generator Server detected an error. Register 2 at ABEND points to the VisualAge Generator Server control block. Register 4 points to a linked list of messages formatted as shown in Figure 43 on page 228.

Table 43. Registers when a SNAP dump is taken or a VisualAge Generator Server ABEND occurs.

Reg.	Value
2	Points to VisualAge Generator Server control block. At offset 272 (hexadecimal offset 110) from the start of the VisualAge Generator Server control block is the address of the initial program profile block, which provides information about the first generated program that was started. At offset 276 (hexadecimal offset 114) from the start of the VisualAge Generator Server control block is the address of the current program profile block, which provides information about the VisualAge Generator program that was running at the time of the abend.
4	Points to the message buffer that contains all messages.

The following diagram shows the format of the message buffer that contains all the messages in the dump.

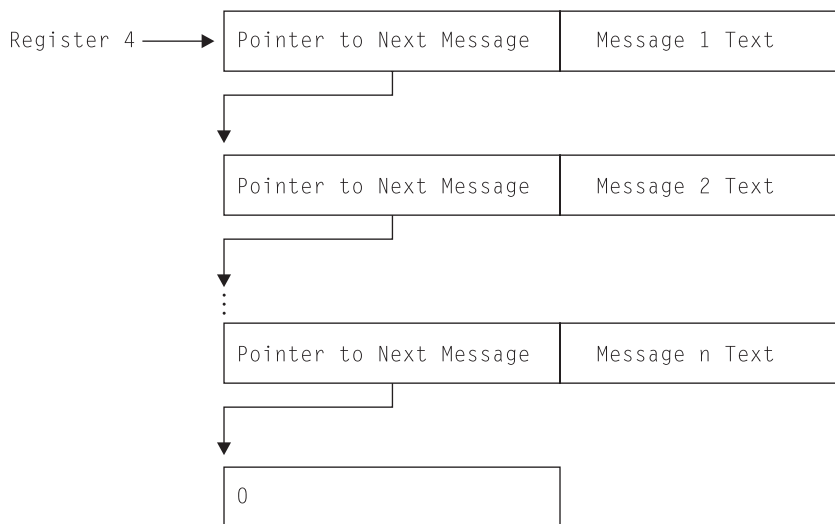


Figure 43. Message Buffer Format

COBOL or Subsystem ABEND Dumps

If the dump is not a VisualAge Generator Server abend, you can use the following method to locate the VisualAge Generator Server control block:

- On MVS CICS systems, locate the MVS CICS Task Work Area (TWA) in the dump. Locate the string *EZERTS-CONTROL* in the TWA. This string is the identifier at the start of the VisualAge Generator Server control block. The * and - characters might be converted to periods in a formatted dump.
- On other systems, locate the string ELARHAPP followed immediately by a program name. ELARHAPP is the identifier at the start of a program profile

block. The four byte address at hex offset 20 in the program profile is the VisualAge Generator Server control block address. If 0, the program might not yet be activated. Do a search for another ELARHAPP control block followed by a program name. See Table 45 for information in the program profile block.

Information in the VisualAge Generator Server Control Block

The following information is in the VisualAge Generator Server control block:

Table 44. Information in the VisualAge Generator Server Control Block

Offset in hex	Length in bytes	Contents
0	16	Control block identifier - *EZERTS-CONTROL*
104	4	MVS CICS EIB Pointer
110	4	Program profile address for current program
114	4	Program profile address for initial program
118	8	Terminal identifier
120	8	User identifier
128	8	Transaction identifier
150	12	EZEDLPSB
1CC	18	Current process or group

Information in an Application, Print Services, or Table Profile Block

Each generated COBOL program contains a profile control block in COBOL working storage initialized with information about the program. The first eight bytes contain an eye-catcher constant identifying whether the program is an application, print services program, or table program. The second eight bytes contain the program name. Other information in the profile block is shown in the following table:

Table 45. Locator Format for Generated COBOL Program Dumps

Offset in hex	Length in hex	Contents
00	08	Program type identifier: ELARHAPP — Application program ELAAHMGC — Print services program ELARMTTP — Table program
08	08	Program name

Table 45. Locator Format for Generated COBOL Program Dumps (continued)

Offset in hex	Length in hex	Contents
10	08	Program generation date (MM/DD/YY)
18	08	Program generation time (HH:MM:SS)
20	04	VisualAge Generator Server control block address
24	02	Generator version
26	02	Generator release
28	02	Generator modification level
2A	10	Reserved
34	08	Target runtime system

How to Find the Current Position in a Program at Time of Error

The VisualAge Generator Server control block identifies the currently running program and process or group at the time of the error (Table 44 on page 229). Associated error messages identify the VisualAge Generator statement number for errors detected by VisualAge Generator Server that need statement identification to resolve the problem. For performance reasons, the generated COBOL program does not keep track of the VisualAge Generator statement number for each generated statement. If a program exception occurs in a generated program, you can determine the VisualAge Generator statement number by finding the COBOL statement that was not successful in a COBOL program listing that contains the VisualAge Generator statements generated as comments.

VisualAge Generator Server Trace Facility

The VisualAge Generator Server trace facility can be used by the IBM Support Center to aid in problem determination, or by the program user to trace program activity.

There are two levels of tracing available:

- VisualAge Generator program source-level tracing
- VisualAge Generator Server runtime-level tracing

With source-level tracing, you can request traces of VisualAge Generator statements, traces of the data, and error codes after every SQL call in a program, except SQL calls made with the SQLEXEC process option. Source-level tracing is enabled with the use of the /TRACE generation option.

Source-level tracing is activated in the runtime environment by specifying trace filter criteria. See “Activating a Trace” on page 232 for more information on activating traces.

With runtime-level tracing, you can request a data stream trace, a VisualAge Generator Server internal dump trace, or a service routine trace. Runtime-level tracing does not require the use of a generation option. Runtime-level tracing is activated in runtime environment by specifying trace filter criteria. See “Activating a Trace” on page 232 for more information on activating traces.

Use these functions only with the assistance of the IBM Support Center. If you use these functions without assistance, large amounts of trace output might be produced based on trace option selection.

Enabling VisualAge Generator Program Source-Level Tracing with a Generation Option

You must specify the /TRACE generation option in order to get source-level trace output.

The VisualAge Generator Developer preprocessor validates the generation option and its parameters. VisualAge Generator Developer creates the necessary COBOL code to accomplish the type of tracing that you request. The /TRACE generation option can be specified with these parameters:

NONE	VisualAge Generator program source-level tracing is not enabled NONE is the default option. /TRACE=NONE excludes any other /TRACE generation options.
SQLERR	Enables tracing of only the SQL error codes returned in the SQL Communications Area (SQLCA) after an SQL call
SQLIO	Enables tracing of both the data and the SQL error codes returned in the SQL Communications Area (SQLCA) after an SQL call Once a program has been generated with the /TRACE=SQLIO generation option, you can then turn on the SQLIO trace option, the SQLERR trace option, or both, or neither.
STMT	Enables tracing of each VisualAge Generator statement that is encountered during the running of the program

Note: More than one parameter can be specified in the /TRACE generation option. The parameter must be separated by comma. The following example shows both SQLIO and STMT are traced:

/TRACE=SQLIO,STMT

Refer to the *VisualAge Generator Generation Guide* for more information on generation options.

Activating a Trace

Tracing is activated during run time either by using the ELAZ transaction in the MVS CICS or IMS/VS environments, or by specifying the ELATRACE DD name in the runtime JCL the MVS/TSO, MVS batch, or IMS BMP environments.

Activating a Trace Session for MVS CICS or IMS/VS

A utility is supplied to activate tracing in the MVS CICS or IMS/VS environments. To start the utility, enter the utility transaction code, ELAZ. The utility transaction must start prior to running the transaction to be traced.

The ELAZ transaction must run in the same region as the transactions to be traced. In IMS, a message processing region can be altered to handle a unique class and the ELAZ transaction and the transactions to be traced set to this class, in order to ensure that the transactions run in the same region. In MVS CICS, enter the ELAZ transaction and the transaction to be traced from terminals attached to the same MVS CICS region.

Figure 44 on page 233 shows the initial panel for the ELAZ transaction that enables you to specify which transactions are to be traced. You use a secondary panel to specify filter criteria for a specific transaction that control what information is traced for that transaction.

Note: For IMS/VS, specify the name of the initial program instead of the initial transaction.

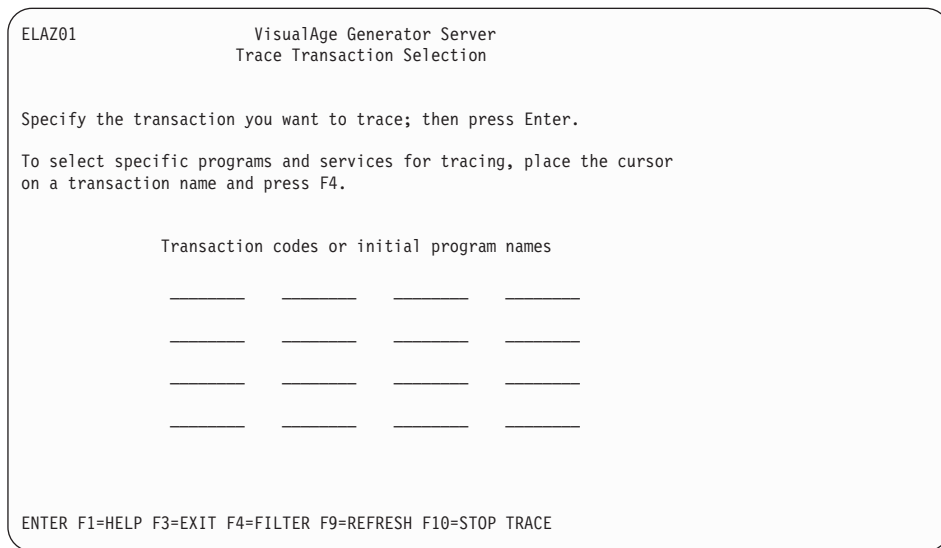


Figure 44. VisualAge Generator Server Trace Transaction Selection Panel

VisualAge Generator Server then presents the panel shown in Figure 45 for trace filter selection:.

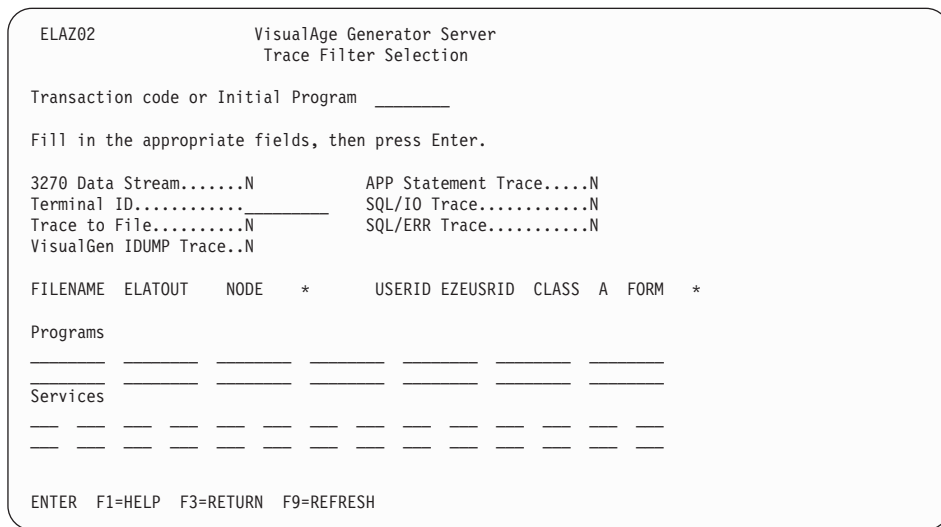


Figure 45. VisualAge Generator Server Trace Filter Selection Panel

The filter criteria include the following:

3270 Data Stream (Y or N)

Specifies whether to trace 3270 data streams

If yes (Y), the 3270 data streams built or received by VisualAge Generator Server are traced. The default is no (N). For IMS/VS environments, 3270 Data Stream Trace option is not allowed.

Terminal ID

Specifies a terminal identifier

If specified, only transactions initiated from that terminal are traced. If not specified, service requests from any terminal are traced.

Trace to File (Y or N)

Specifies whether the trace output goes to a file

If yes (Y), the trace output of VisualAge Generator Server is sent to the ELAT transient data queue in MVS CICS and to an IMS/VS message queue for transaction ELATOUT in IMS/VS. This trace is also written to an in-storage wrap-around trace buffer.

If no (N) the trace output goes to an in-storage wrap-around trace buffer. The size of this trace buffer is defined during customization of VisualAge Generator Server.

Y must be specified if you specify Y (yes) for the SQL/IO Trace or SQL/ERR Trace options. All trace output for SQL/IO and error tracing is sent to a file, not to the in-storage wrap-around trace buffer.

Note: You cannot trace to file if the tracing transaction uses the modifiable express PCB because VisualAge Generator Server uses the modifiable express TP PCB to write to the message queue. Unpredictable results can occur.

VisualGen IDUMP Trace (Y or N)

Specifies whether to dump VisualAge Generator Server internal storage areas

If yes (Y), the trace facility provides dumps of certain VisualAge Generator Server internal storage areas. The default is no (N), no internal storage dumps.

APP Statement Trace (Y or N)

Specifies whether to trace VisualAge Generator statements in a program

If yes (Y), the trace facility provides the process name and the statement for each VisualAge Generator statement that the program processes. You must use the /TRACE=STMT generation option to enable this type of tracing. The default is no (N).

SQL/IO Trace (Y or N)

Specifies whether to trace SQL/IO

If yes (Y), the trace facility provides traces of the data and error codes on the return from the SQL call. The VisualAge Generator SQL process name, the process option, the process object, the SQL function name, and the VisualAge Generator data item name, length, type, and contents are given. You must use the /TRACE=SQLIO generation option to enable this type of tracing. The default is no (N).

SQL/ERR Trace (Y or N)

Specifies whether to trace SQL error information

If yes (Y), the trace facility provides traces of the error information that comes back from SQL on every database call. The SQLCODE, SQLERRP, SQLSTATE, SQLWARN, SQLERRD, SQLEXT, and SQLERRMC codes are given. You must use the /TRACE=SQLERR or the /TRACE=SQLIO generation option to enable this type of tracing. The default is no (N).

FILENAME

The system resource name for the trace output. The default is ELATOUT.

NODE

1 to 8 characters that specify the system node ID. The default is the current system node ID.

USERID

1 to 8 characters that specify the user ID. The default is the value of the EZEUSRID special function word.

CLASS

A single character that specifies the print class. The default is A.

FORM

1 to 4 characters that specify the form number for print output. The default is your location's standard form.

Applications

Specifies whether to limit the trace to certain application programs or print services programs

If specified, only the requested programs are traced.

Services

Specifies whether to limit the trace to certain services

If specified, only the requested services are traced. Otherwise all service numbers are traced if the other criteria are met.

Note: The entry to ELARSINI (initialization service) and the exit from ELARSTRM (cleanup service) are not traced. ELARSINI initializes the trace facility. ELARSTRM ends the trace facility.

If you are running a trace to aid in problem determination, enter the filter criteria as directed by the IBM support center.

Activating a Trace Session for MVS/TSO, MVS Batch, or IMS BMP

Tracing is activated by providing trace filters in a preallocated data set with the DD name ELATRACE before starting the program or job to be traced. ELATRACE contains control statements which control the programs and events to be traced. The attributes for the data set are LRECL=80, DSORG=PS, and RECFM=FB. If the ELATRACE data set is empty or allocated as DD DUMMY, all services are traced, data streams are not traced, and SQL I/O, SQL errors, and program statement are not traced even if enabled through a /TRACE generation option. Figure 46 shows the correct syntax for the trace control statements.

```
:FILTER DATASTREAM=Y|N
:FILTER TRACETOFILE=Y|N
:FILTER APPSTMT=Y|N
:FILTER SQLIO=Y|N
:FILTER SQLERR=Y|N
:FILTER IDUMP=Y|N
:APPLS

:
[ name ]

:
: EAPPLS
: SERVICES

:
[ service number ]

:
: ESERVICES
: EFILTER
```

Figure 46. ELATRACE Data Set Entries

Notes:

1. Only one program name or service number can be entered on each line.
2. The :FILTER and :EFILTER tags are required if any other tags are included in the ELATRACE data set.

3. More than one filter can be specified on a line. The filters must be separated by 0 or more blanks. The example below shows sample :FILTER statements that are valid and equivalent:

```
:FILTER APPSTMT=Y
:FILTER SQLERR=Y

:FILTER APPSTMT=YSQLERR=Y

:FILTER APPSTMT=Y SQLERR=Y

:FILTER APPSTMT=Y           SQLERR=Y
```

The filters cannot be continued on the next line. The statement shown below is not valid:

```
:FILTER APPSTMT=Y SQLERR=
      Y
```

The control card tags and attributes that control filtering have the following meaning:

- :FILTER** Options controlling what information is traced and where trace data is written
- The following attributes can be used with the :FILTER statement:
- DATASTREAM=Y | N
If DATASTREAM=Y is specified, the 3270 data streams built or received by VisualAge Generator Server are traced. The default value is N, no data stream tracing.
 - TRACETOFILE=Y | N
If TRACETOFILE=Y is specified, the trace output is directed to the preallocated data set named ELATOUT in addition to being directed to an in-storage wrap-around trace buffer.
If TRACETOFILE=N is specified, the trace output goes to an in-storage wrap-around trace buffer. The size of this trace buffer is defined during customization of VisualAge Generator Server. The default for the TRACETOFILE option is N.
TRACETOFILE=Y must be specified if SQLIO=Y or SQLERR=Y are specified. All trace output for SQL I/O and SQL errors is directed to the ELATOUT data set, not to the in-storage wrap-around trace buffer.
 - APPSTMT=Y | N
If APPSTMT=Y is specified, the trace facility provides the process name and the statement for each VisualAge Generator statement that the program processes. You must

use the /TRACE=STMT generation option to enable this type of tracing. The default for the APPSTMT option is N.

- SQLIO=Y|N

If SQLIO=Y is specified, the trace facility provides traces of the data and error codes on the return from the SQL call. The VisualAge Generator SQL process name, the process option, the process object, the SQL function name, and the VisualAge Generator data item name, length, type, and contents are given. You must use the /TRACE=SQLIO generation option to enable this type of tracing. The default for the SQLIO option is N.

- SQLERR=Y|N

If SQLERR=Y is specified, the trace facility provides traces of the error information that comes back from SQL on every database call. The SQLCODE, SQLERRP, SQLSTATE, SQLWARN, SQLERRD, SQLEXT, and SQLERRMC codes are given. You must use the /TRACE=SQLERR or the /TRACE=SQLIO generation option to enable this type of tracing. The default for the SQLERR option is N.

- IDUMP=Y|N

If IDUMP=Y is specified, the trace facility provides dumps of certain VisualAge Generator Server internal storage areas. The default for the IDUMP option is N, no internal storage dumps.

:APPLS

Application program names or print service program names

If program names are specified, only the specified programs are traced. Otherwise service requests from each generated program are traced. Up to 16 program names can be specified.

:SERVICES

Service numbers

If service numbers are specified, only those specific services are traced. To trace all service numbers, 999 must be specified. Otherwise, up to 32 service numbers can be specified.

Note: The entry to ELARSINI (initialization service) and the exit from ELARSTRM (cleanup service) are not traced. ELARSINI initializes the trace facility. ELARSTRM ends the trace facility.

Deactivating a Trace Session

To stop all trace activity for MVS CICS or IMS/VS transactions, use the ELAZ transaction to delete the transaction codes from the list of transactions to be

traced by using the F10 function key. When a transaction ends and is subsequently restarted, tracing does not start if the transaction code no longer appears in the transaction list.

To stop tracing in MVS/TSO, MVS batch, or IMS BMP, cancel the program and remove the ELATRACE and ELATOUT DD cards from the runtime JCL.

Printing Trace Output

If the trace output is not directed to a file for the MVS CICS or IMS/VS environments, or the ELATOUT data set is not allocated for MVS/TSO, MVS batch, or IMS BMP jobs, trace output is written to a wrap-around trace buffer in memory. The trace output can be seen in dumps taken when programs end abnormally.

Printing the Trace Output in MVS CICS

Trace output for MVS CICS is routed to an extrapartition transient data queue which is directed to a data set named ELATOUT if you direct the trace output to a file by specifying yes (Y) on the ELAZ02 panel. The ELATOUT data set has the attributes of LRECL=133, RECFM=FBA. The file can be printed as directed on the DD statement for ELATOUT in the CICS startup JCL.

Printing the Trace Output in MVS/TSO, MVS Batch, and IMS BMP

Trace output is directed to the ELATOUT DD statement and is printed as directed on the DD statement.

Printing the Trace Output in IMS/VS

The trace entries written to an IMS message queue can be printed with the ELAEPUTL utility. The sample job stream shipped with VisualAge Generator Server is run in order to drain and print the trace output. The job stream must be tailored to use the IN=ELATOUT parameter on the EXEC IMSBATCH statement.

```

//*****
//** ELAMQJUD - JCL TO DRAIN AND PRINT THE ELADIAG MESSAGE QUEUE
//**          FOR VISUALAGE GENERATOR SERVER.
//**          THIS PROGRAM RUNS AS A BMP.
//**
//** LICENSED MATERIALS - PROPERTY OF IBM
//** 5648-B02 (C) COPYRIGHT IBM CORP. 1994, 1998
//** SEE COPYRIGHT INSTRUCTIONS
//**
//** STATUS = VERSION 1, RELEASE 2, LEVEL 0
//**
//** TO TAILOR THIS JOBSTREAM:
//**     1. COPY A JOBCARD.
//**     2. CHANGE IN= TO THE NAME OF YOUR ERROR DIAGNOSTIC
//**        QUEUE.
//**     3. MAKE SURE THAT THE TRANSACTION SPECIFIED BY IN=
//**        AND THE ELAMPUTL PROGRAM ARE STARTED BY IMS.
//**
//** RETURN CODES
//**     0 - SUCCESSFUL COMPLETION
//**     4 - NO MESSAGES ON QUEUE TO DRAIN.
//**    16 - FATAL ERROR. PROCESSING TERMINATED
//**    20 - OPEN FAILED ON ELAPRINT
//**
//*****
//DRAINMQ    EXEC IMSBATCH,MBR=ELAEPUTL,
//           PSB=ELAMPUTL,IN=ELADIAG,RGN=4096K
//G.STEPLIB  DD
//           DD
//           DD DSN=CEE.SCEERUN,DISP=SHR
//           DD DSN=ELA.V1R2M0.SELALMD,DISP=SHR
//G.ELAPRINT DD SYSOUT=*
//G.SYSOUT   DD SYSOUT=*
//G.SYSPRINT DD SYSOUT=*
/*

```

Figure 47. Sample job to drain and print ELAT

Reporting Problems for VisualAge Generator Server

VisualAge Generator provides a set of manuals that contain information that can assist you in using the VisualAge Generator products. Check these manuals first for answers to your problems or questions. A list of the VisualAge Generator manuals are located in the Preface section of each manual.

Documenting information is vital not only in locating the source of the problem, but also in resolving the problem.

Gather the necessary documented information before you contact the IBM Support Center to report your problem. The following list outlines the minimum required documentation.

- The release level of VisualAge Generator Server and the operating system it is running under
- The list of all PTFs and additional APARs that have been applied to VisualAge Generator Server
- All VisualAge Generator Server and non-VisualAge Generator Server error messages including text and return codes
- The first indication that the problem has occurred
- A detailed description of the problem and any additional information attributing to the problem
- Determine whether the problem occurred during generation, compilation, or run time
- Can the problem be recreated?
- Does the program run successfully under VisualAge Generator Developer?
- Do other VisualAge Generator generated programs fail? If not, does the problem program do something unique that the programs that work do not do?
- Does the program interface with non-VisualAge Generator programs? If so, what language are the non-VisualAge Generator programs written in?
- Any recent changes made to VisualAge Generator Server, the program, MVS, DB2, DL/I or other related software
- The terminal types involved if the problem is terminal related
- Languages installed and the one used to create the problem again
- For the IMS environment:
 - Is the program to run as BMP or MPP?
 - If MPP, is the program conversational?
 - If BMP, is it batch or message driven?
 - Is the work database (ELAWORK) DL/I or DB2?
- For the MVS CICS environment:
 - Is the program to run as segmented or nonsegmented?
 - What diagnostic controller options are being used?

Your IBM Support Center representative might request documentation depending on the type of problem. This documentation might include the following:

- The program with associates exported in external source format
- The generated COBOL source code
- The generation options

- JCL output from preparation or run time
- DB2 information (bind results, table definitions, and so forth)
- IMS definition from stage 1 (APPLCTN, TRANSACT, and so forth)
- MVS CICS table entry information (PCT, PPT, FCT, and so forth)
- Dump of the error

These files can be put on tape or cartridge and sent to the support center. Please follow these instructions when creating the tape or cartridge:

1. Create all tapes (reels or cartridges) as Standard Label (SL) NOT Non Label (NL) or Bypass Label Processing (BLP).
2. Use ONLY IBM standard utilities, such as IEBGENER, IEBCOPY, or IDCAMS.
3. On an ESA system there is a DCB parameter to compact data on the tape. The parameter is coded:

TRTCH=NOCOMP or TRTCH=COMP

Do not use the COMP parameter. Instead specify: TRTCH=NOCOMP

4. Tell the IBM Support Center representative if the dump was taken from an XA or ESA system. Also, send the dump already formatted.
5. Provide a hard copy of the JCL used to create the tape.
6. Any data that you supply should be marked with your:
 - Account name
 - Problem number
 - APAR number (if assigned)

Problem Reporting Sequence

When you report a problem, tell the IBM Support Center representative the symptom string you received if an ABEND occurred. A symptom string contains an ABEND code, the name of the VisualAge Generator Server module in control at the time of the error, and the program ID.

Figure 48 on page 243 shows the sequence of events that should occur after you describe the problem to the IBM Support Center.

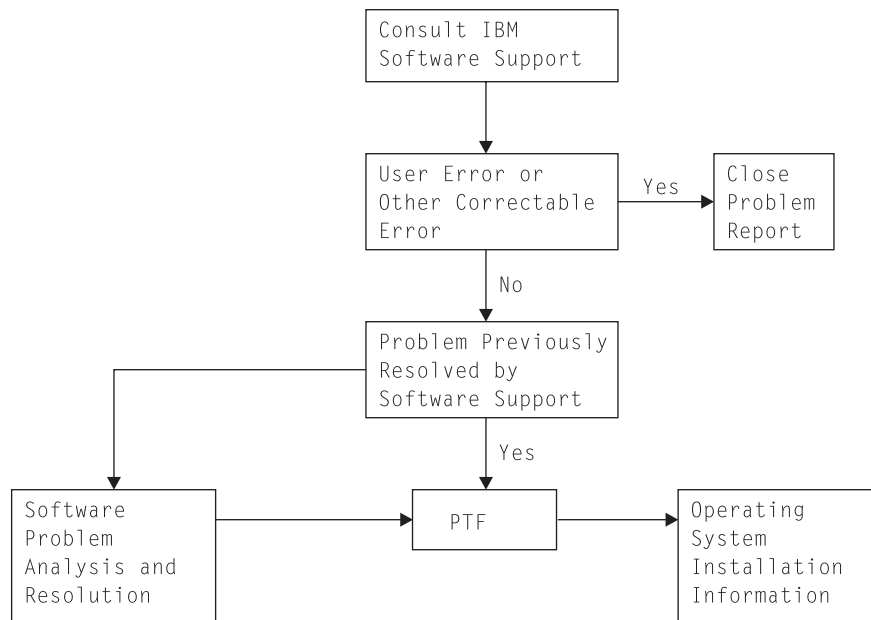


Figure 48. Problem Reporting Sequence

If the IBM Support Center has an existing solution for the problem, see the appropriate system documentation for instructions on applying the patch (if the resolution is a temporary patch) or program temporary fix (PTF).

If the IBM Support Center cannot resolve the problem, or a PTF is unavailable, an Authorized Program Analysis Report (APAR) number is assigned to your problem. You might be asked to assist in collecting information to submit an APAR.

Chapter 26. Common Messages during Preparation for MVS Systems

This chapter contains some error messages from other products. It is not a complete list. For a complete explanation of product messages, refer to the documentation provided with that product.

Common Abend Codes during Preparation

Only the most frequently occurring preparation abend codes are listed in this section. If you receive any other abend code or need a more complete explanation of one of the abend codes, refer to the documentation for that product.

System B37

If there are a large number of maps in a particular map group causing a large map group format module to be produced during COBOL generation, the temporary work space is filling up. This can be corrected by increasing the size specified on the WSPC parameter in the proc ELAL.

System 213, or System 230

Two program developers tried to update the directory of a PDS at the same time. Submit the job again.

IMS 3022

The map group that was generated into MFS source resulted in one or more MFS control blocks that exceeded the 32,748-byte limit. The map group cannot be processed by MFS in its current form. Use the map definition facility to split the map group into two or more separate map groups, and then change your program as necessary.

MFS Generation Messages

Only the most frequently occurring MFS generation messages are listed in this section. If you receive other error messages that start with DFS or if you need a more complete explanation of one of the messages, refer to the IMS messages and codes manual for your release of IMS.

DFS1141I **name FMT DOES NOT DEFINE
DEVICE INPUT DESCRIPTION
FOR INPUT MESSAGE
DESCRIPTION, FMT NOT
PROCESSED**

Explanation: This message can occur when a map group was originally defined and generated with both terminal and printer maps. Then the map group was changed to have only printer maps and generated again. This leaves a part in the IMS REFERRAL library for the terminal maps and causes MFS assemblies to end with errors.

User Response: Run the MFSRVC procedure that is supplied with IMS and specify the SCRATCH function to remove this part from the IMS REFERRAL library. Refer to the MFS utilities documentation for your release of IMS for additional information.

DFS1162I **xxxxxxx WARNING: ATTR=nn
SPECIFIED FOR DFLDNAME
WHICH HAD NO EATTR=
SPECIFICATION.**

Explanation: You specified the /NOMFSEATTR generation option or included the NOEATTR parameter for one or more devices in the /MFSDEV generation option.

User Response: None, provided you wanted to specify devices that do not support extended attributes.

DFS1428I **SC=08 LTH=NN,NN EXCEEDS 4
SIGNIFICANT DIGITS. LAST 4
DIGITS USED.**

Explanation: This message occurs if a map contains a variable field longer than 8000 bytes for a print map or longer than 1 less than the display size for a terminal map.

User Response: Use map definition to split the variable field into smaller fields. Change your program to use the smaller fields and then generate the map group and program again.

DFS1587I **SC=04 EGCS FIELD SPECIFIED
ON AN EVEN COLUMN**

Explanation: You defined a DBCS constant or variable field that starts on an even column (in other words, the data starts in an even column).

User Response: If the device you are using is an IBM Personal System/55* or is in the IBM 5550 family, you can ignore this message. Otherwise, use map definition to change the definition of the map.

DB2 Precompiler and Bind Messages

Only the most frequently occurring DB2 precompiler and bind messages are listed in this section. If you receive other messages that start with DSN or if you need a more complete explanation of one of the messages, refer to the documentation for your release of DB2.

DSNH039I **S PRECOMPILED INTERNAL
LIMIT EXCEEDED**

Explanation: A limit for the DB2 precompiler has been exceeded. This can occur in programs that contain a large number of SQL process options.

User Response: Split the program into smaller programs. If you use a DXFR or a CALL statement to split the program, you need to bind the resulting programs together. In addition, if

you use an XFER statement to split the programs and are generating the programs for MVS/TSO, MVS Batch, or IMS BMP, you need to bind the resulting programs together.

DSNX100I **BIND SQL WARNING**

Explanation: One or more DB2 tables have not been created. The tables that do not exist will be identified in an explanation associated with the message by:

xxxxxxx IS NOT DEFINED
where xxxxxxx is the table name.

User Response: Create the necessary DB2 tables and synonyms.

DB2/VSE Precompiler Messages

Only the most frequently occurring DB2/VSE precompiler messages are listed in this section. If you receive other messages that start with ARI or if you need a more complete explanation of one of the messages, refer to the documentation for your release of DB2/VSE.

ARI0501I AN SQL WARNING HAS OCCURRED. SQLCODE DESCRIPTIVE TEXT.

Explanation: This message is followed by message ARI505I, which provides the SQLCODE and ROWCOUNT information. The SQLCODE descriptive text describes the SQLCODE error condition. If SQLCODE is 204, one or more SQL tables have not been created. The tables that do not exist are identified in an explanation associated with the message by:

xxxxxxx was not found in the system catalog where xxxxxxx is the table name.

User Response: Create the necessary SQL tables and synonyms before the running the program.

ARI054E SQL STATEMENT CONTAINS MORE THAN 256 HOST VARIABLES.

Explanation: A limit of the host variable reference in a single SQL statement has been exceeded.

User Response: Change the SQL statement to contain fewer than 256 host variables.

ARI0565I THE INPUT SOURCE CONTAINS EITHER NO SQL STATEMENTS, OR NO SQL STATEMENTS THAT WILL CREATE A PACKAGE.

Explanation: This can occur if one of the following is true:

- The program contains no SQL process options.

- The program contains only CALL EZECONCT, EZEROLLB, RESET or EZECOMIT statements. These calls result in CONNECT, ROLLBACK or COMMIT SQL statements that do not create a package.

User Response: The preprocessing continues. The generated program can be run successfully. If you have SQL process options that result in SQL calls other than CONNECT, ROLLBACK, or COMMIT in your program, then you should not receive this message.

ARI5387E The SQL statement length exceeds 8192 positions

Explanation: A limit of the SQL statement length has been exceeded. The possible causes are the following:

- The SQL statement is too long.
- The end of statement delimiter is missing.
- A FOR UPDATE OF clause is added to the end of a static query statement.
- The expansion of a host structure reference has increased the size of the SQL statement.

User Response: Either break the SQL statement into smaller statements or remove unnecessary conditions that might exist in the statement.

ARI5400I Warning: statement-type is not supported.

Explanation: The SQL statement shown is not supported by the DB2/VSE system. The statement-type can be either DECLARE TABLE, DECLARE STATEMENT, or DISCONNECT.

User Response: If your program depends on the code expansion, change your program so that it does not depend on either of the above

statements. Or, install the release of DB2/VSE that supports this statement-type.

COBOL Compilation Messages

Only the most frequently occurring COBOL compilation messages are listed in this section. If you receive other compilation messages that start with IGY or if you need a more complete explanation of one of the messages, refer to the documentation for your release of COBOL.

IGYPS2015I The paragraph or section prior to paragraph or section EZEMAIN-PROCESS did not contain any statements.

IGYPS2023I Paragraphs prior to section EZEMAIN-PROCESS were not contained in a section

Explanation: These two messages occur if your program has been processed by the DB2 precompiler.

User Response: They are normal messages that you can ignore.

IGYOP3091W
Code from "?" to "?" can never be executed, and was therefore discarded.

IGYOP3093W
The "PERFORM" statement at "?" cannot reach its exit.

IGYOP3094W
There may be a loop from the "PERFORM" statement at "?" to itself. "PERFORM" statement optimization was not attempted.

Explanation: These messages occur if your program has been processed using the OPTIMIZE compiler option.

User Response: These are normal messages that you can ignore.

IGYPA3013W
Data item "?" and "?" had overlapping storage. An overlapping move will occur at execution time.

Explanation: This message occurs if your program attempts to assign the value of a data item to the same data item.

User Response: You might want to check that you really intended to do this.

IGYPG3113W
Truncation of high-order digit positions may occur due to precision of intermediate results exceeding 30.

Explanation: This message might occur if your program was generated with the /MATH=COBOL generation option.

User Response: You might want to change the arithmetic expression identified in the message. For example, you could split the expression into several smaller ones.

If you do not change the expression, ensure that the intermediate values will fall within the precision that COBOL supports. Refer to the programming guide for your release of COBOL for more information about the precision of intermediate results.

IGYSC2025W

"EZEPCB-?" or one of its subordinates was referenced, but "EZEPCB-?" was a "LINKAGE SECTION" item that did not have addressability. This reference will not be resolved successfully at execution.

Explanation: This warning message occurs when PCBs or any data structure is generated in the linkage section, but is not used in a statement.

User Response: Ignore the messages and the program will work correctly.

Chapter 27. Common System Return Codes for MVS Systems

The information within this chapter is diagnosis, modification, or tuning information.

VisualAge Generator Server messages might include return codes from databases or operating systems that are being used. This could include DB2, DL/I, MVS VSAM, or MVS CICS EXEC Interface Block (EIB) codes.

This chapter contains only the most common errors that occur during file input and output operations.

The return codes included in this chapter are for the following databases and operating systems:

- MVS CICS
- MVS/TSO
- DB2
- DL/I
- VSAM
- COBOL

Common EZERT8 Return Codes

The EZERT8 special function word contains a return code indicating a reason that a file I/O process option or a CALL to a service routine is not successful. Codes specific to the system or the access method are returned when the user requests system codes with the /SYSCODES generation option.

If the /NOSYSCODES generation option is specified, the system return codes are converted to VisualAge Generator Server return codes. This allows applications developed previously under Cross System Product to receive the same return codes as before.

System Return Code Formats for EZERT8

The following table shows the formats of EZERT8 by specific environment:

System	Compatibility Considerations
MVS CICS	<p data-bbox="496 302 1234 474">If EZERT8 is in the form RSnnnnnn, look under nnnnnn in “Common System Return Codes in EZERT8” on page 254. Otherwise, the first 2 characters of EZERT8 contain the first byte of the EIBFN from the CICS EXEC interface block (EIB). The remaining 6 characters contain bytes 0-2 of the EIBRCODE, also from the CICS EXEC interface block.</p> <p data-bbox="496 499 1195 617">If the REPLY option is specified on a CALL statement, and the CALL is implemented with the MVS CICS LINK, the first 2 characters of EZERT8 contain the first byte of the EIBFN and the remaining 6 characters contain bytes 0-2 of the EIBRCODE.</p> <p data-bbox="496 642 1110 693">Refer to the CICS application programmers’ guide for an explanation of the EIB codes.</p>

System	Compatibility Considerations
MVS/TSO	<p data-bbox="516 191 1154 247">If EZERT8 is in the form RSnnnnnn, look under nnnnnn in “Common System Return Codes in EZERT8” on page 254.</p> <p data-bbox="516 275 1227 331">SEQ: EZERT8 contains the COBOL status key value or values in the first 2 characters. The remaining 6 characters are zeros.</p> <p data-bbox="516 359 1211 415">SEQRS: The contents of EZERT8 depend on the operation that was not successful:</p> <ul data-bbox="516 432 1247 741" style="list-style-type: none"> <li data-bbox="516 432 1247 541">• If a dynamic allocation is not successful, the first 3 bytes of EZERT8 contain the value S99 (for SVC 99, dynamic allocation), byte 4 is the SVC 99 return code in hexadecimal, and bytes 5-8 contain the error reason code in hexadecimal. <li data-bbox="516 558 1198 615">• If an OPEN is not successful, EZERT8 contains return code 8 ('0000008'). <li data-bbox="516 625 1227 682">• If a READ end-of-file condition occurs, EZERT8 contains return code 4 ('0000004'). <li data-bbox="516 693 1247 741">• If a READ, WRITE, or CLOSE is not successful, EZERT8 contains return code 12 ('0000012'). <p data-bbox="516 768 1252 909">VSAM: EZERT8 contains the COBOL status key value or values in the first 2 characters followed by 2 characters for the COBOL VSAM return code (VSAM feedback code), 1 character for the COBOL VSAM function code (VSAM component code), and 3 characters for the COBOL VSAM feedback code (VSAM reason code).</p> <p data-bbox="516 936 1203 993">VSAMRS: The operation that is not successful determines the contents of EZERT8:</p> <ul data-bbox="516 1010 1260 1423" style="list-style-type: none"> <li data-bbox="516 1010 1247 1119">• If a dynamic allocation is not successful the first 3 bytes of EZERT8 contain the value S99 (for SVC 99, dynamic allocation), byte 4 is the SVC 99 return code in hexadecimal, and bytes 5-8 contain the error reason code in hexadecimal. <li data-bbox="516 1129 1227 1239">• If an OPEN or CLOSE is not successful, the first 2 bytes of EZERT8 contain the error code from the VSAM application control block (ACB) in hexadecimal. The remaining 6 characters are zeros. <li data-bbox="516 1249 1260 1423">• If an operation other than OPEN or CLOSE is not successful, the first 2 characters are zeros followed by 2 characters for the COBOL VSAM return code (VSAM feedback code), 1 character for the COBOL VSAM function code (VSAM component code), and 3 characters for the COBOL VSAM feedback code (VSAM reason code). <p data-bbox="516 1451 1243 1554">For VSAM codes, refer to MVS/Extended Architecture VSAM Administration: Macro Instruction Reference. For SVC 99 codes, refer to MVS/Extended Architecture System Programming Library: System Macros and Facilities.</p>

System	Compatibility Considerations
MVS Batch	<p>If EZERT8 is in the form RSnnnnnn, look under nnnnnn in “Common System Return Codes in EZERT8”.</p> <p>GSAM: EZERT8 contains the DL/I status code after an I/O process. The last 6 characters of EZERT8 are blanks.</p> <p>Otherwise, same as MVS/TSO in this table.</p>
IMS/VS	<p>The only files that can be used in this environment are serial files associated with IMS message queues. EZERT8 contains the DL/I status code after an I/O process to one of these files. The last 6 characters of EZERT8 are blanks.</p>
IMS BMP	<p>GSAM: EZERT8 contains the DL/I status code after an I/O process. The last 6 characters of EZERT8 are blanks.</p> <p>IMS message queue: EZERT8 contains the DL/I status code after an I/O process. The last 6 characters of EZERT8 are blanks.</p> <p>Otherwise, same as MVS/TSO in this table.</p>

Common System Return Codes in EZERT8

The following table gives an explanation of the most common values that you receive in EZERT8 when system codes are requested. If your return code is not listed here, or you would like more information, refer to the table in “System Return Code Formats for EZERT8” on page 252 and then the appropriate manuals for your environment.

Table 46. EZERT8 return codes.

System	Return code	Meaning
MVS/TSO	A0000000	VSAM open error - empty indexed file
MVS/TSO	BC000000	VSAM open error - file is not in VSAM format
MVS/TSO	S9940210	File not available
MVS/TSO	S9940440	File not found
MVS/TSO	S99417**	File not found
MVS/TSO	00000004 on non-VSAM file	End of file
MVS/TSO	00000008 on non-VSAM file	Error opening file
MVS/TSO	00000012 on non-VSAM file	Error on I/O or closing a file
MVS/TSO	0008*004 for nonrelative	End of file
MVS/TSO	0008*004 for relative	No record found
MVS/TSO	0008*008 for an ADD request	Duplicate record
MVS/TSO	0008*016 if scanning an indexed record	End of file

Table 46. EZERT8 return codes. (continued)

System	Return code	Meaning
MVS/TSO	0008*016 if not scanning an indexed record	No record found
MVS/TSO	0008*028	File full
MVS/TSO	0008*116	No record found
MVS/TSO	*****74	No record found
Note: * represents any character.		
Note: MVS/TSO in this table includes MVS Batch and BMP.		
MVS CICS	ffrrrrr Remote CALL or CREATX	Other MVS CICS errors: <ul style="list-style-type: none"> • ff = Hexadecimal representation of EIBFN byte 0 • rrrrrr = Hexadecimal representation of EIBRCODE bytes 0-2
MVS CICS	0A010000 on scanning a temporary storage queue	End of file
MVS CICS	0A010000 on direct I/O to a temporary storage queue	No record found
MVS CICS	0A080000 on temporary storage queue	File is full
MVS CICS	060F0000 on VSAM file	End of file
MVS CICS	00000000 Remote CALL or CREATX	Successful
MVS CICS	00000203 Remote CREATX	Transaction identifier that is not valid
MVS CICS	00000204 Remote CALL	Program name that is not valid
MVS CICS	00000207 Remote CALL or CREATX	System identifier that is not valid
MVS CICS	00000208 Remote CALL	Link out of service that is not valid
MVS CICS	06810000 on VSAM file	No record found
MVS CICS	06820000 on VSAM file	Duplicate record
MVS CICS	06830000 on VSAM file	File is full
MVS CICS	08E10000 on transient data	Format error
MVS CICS	08010000 on a transient data queue	End of file
MVS CICS	08020000 on a transient data queue	File not found
MVS CICS	08080000 on transient data	Transient data queue not open
MVS CICS	12320000	Queue is already in use

VisualAge Generator Server Return Codes

The return codes list is sequenced by return code, with the alphabetic return codes (A to Z) occurring before the numeric return codes (0 to 9). If you specified /NOSYS_CODES, EZERT8 will contain return codes that are compatible with the Cross System Product codes.

Table 47. VisualAge Generator Server Return Codes

RC	Component	Probable Cause
Cnn	File control/request	These return codes do not have a VisualAge Generator mnemonic. Either MVS CICS returned an IOERR error (if MVS CICS), or VSAM returned a return code of 12 on file input/output. The nn is replaced by the VSAM reason code from the feedback field. For more information, see the <i>MVS VSAM Administration Manual: Macro Instruction Reference</i> manual.
Fnn	File control/request	These return codes are CICS EIBRCODES, other than ILLOGIC, IOERR, and those that have VisualAge Generator mnemonics. The nn is replaced by the EIBRCODE (byte 0). For more information, see the <i>CICS/VS Application Programmer's Reference Manual (Command Level)</i> Note: All return codes, other than the ones that have VisualAge Generator mnemonics, cause the program to end. An error message is issued to inform you that the program ended because of a send/receive error. The error message includes the return code.
FE1	File Control/request	Transient data queue - Queue length and VisualAge Generator record length do not match. The VisualAge Generator mnemonic FMT is set.
F02	File Control/request	Transient data queue - File not found. VisualAge Generator mnemonic FNF is set.
F08	File control/request	An attempt was made to gain access to an extrapartition transient data queue, but the queue has not been opened yet. Exit and use CEMT to open the queue.

Table 47. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
02	EZCHART	An incorrect number of parameters was passed on a call to EZCHART.
03	EZCHART	GETMAIN error - An attempt to obtain storage for data conversions was not successful.
04	EZCHART	A data overflow occurred during the conversion of numeric data to floating point format. Data must fit into 6-place floating point format.
06	EZCHART	The printer depth was not specified in Chart Control.
07	EZCHART	The printer width was not specified in Chart Control.
08	EZCHART	The number of print copies was not specified in Chart Control.
09	EZCHART	The printer name was not specified in Chart Control.
10	EZCHART	A load or link error occurred for a GDDM* routine.
52	Terminal support	You attempted to enter VisualAge Generator on an unsupported device (such as a 3278-52 terminal). This device is not supported by VisualAge Generator.
101	Message processing	The message was truncated.
102	Contents control	The module specified on a LOAD macro is already in storage.
	File control/request	The end of file was reached. The VisualAge Generator mnemonic EOF is set. Note: EOF is set when a SCAN is performed on an empty file.
	Service request	An ITEMERR condition was received from MVS CICS because the maximum number of records allowed in a temporary storage queue (32767) was exceeded.

Table 47. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
103	File control/request	You performed an operation on a record that has a duplicate key, or a key in the record for an alternate index is duplicated. The I/O operation completed, and the VisualAge Generator mnemonic DUP is set.
104	File control/request	The end of file was reached. The VisualAge Generator mnemonic EOF is set.
115	Service request	An EXEC CICS ENQ was not successful.
116	Service request	An EXEC CICS DEQ was not successful.
20B	Storage allocation	Operands that are not valid were specified on either a GETMAIN or FREEMAIN macro.
20C	Defined data set	The data set name specified on an issued DEFDS command already exists in the external work file.
	Storage allocation	An error occurred while processing a FREEMAIN macro.
200	Service request	An service request was issued that is not valid. This is a system error. Contact the IBM Support Center.
201	File open/connect	The connection already exists. The possible cause is a file with the same name is already in use. VisualAge Generator mnemonic FNA is set.
	Message processing	Variables were passed to be built into the message, but the message contained no variable fields; or, the message contained variable fields, and no variables were passed for them.
201- 206	Service request	Service request errors occurred while processing a DEFDS command. This is a system error. Contact the IBM Support Center.
202	File control/request	Record not found. The VisualAge Generator mnemonic NRF is set.

Table 47. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
	Storage allocation	The ORIGIN specified on a FREEMAIN macro does not match storage already in use.
203	File control/request	The record was not found. The VisualAge Generator mnemonic NRF is set.
	Storage allocation	Either the ORIGIN specified on a FREEMAIN macro does not begin on a doubleword boundary, or 0 LENGTH was specified on a GETMAIN.
204	Storage allocation	An attempt has been made to free storage that has not been allocated or that has already been freed.
205	File control/request	The record was not found. The VisualAge Generator mnemonic NRF is set. Note: The VisualAge Generator mnemonic NRF is set when a SCAN or SCANBACK is preceded by a SET record SCAN on an empty indexed file.
	Storage allocation	The LENGTH specified on a FREEMAIN macro is 0.
206	File control/request	You attempted to store a record with a duplicate key while using an index that does not allow duplicate keys. The VisualAge Generator mnemonics UNQ and DUP are set.
207	File control/request	The record was not found. The VisualAge Generator mnemonic NRF is set.
208	File control/request	An error occurred when you attempted to connect or write to the log file on MVS CICS. A possible reason for the error is that no entry was found for the log file in the destination control table (DCT).
	Service request	The NDSNAME in an ALTDS request is not valid. This is a system error. Contact the IBM Support Center.
	Storage allocation	The storage specified on a FREEMAIN macro is already free.

Table 47. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
209	Service request	The name specified by the NDSNAME on an ALTDS command already exists in the external work file. This is a system error. Contact the IBM Support Center.
210- 211	Service request	Miscellaneous errors occurred on an ALTDS request. This is a system error. Contact the IBM Support Center.
212	Service request	An I/O error occurred while copying data from the work file to the external work file during an ALTDS service request.
213	Service request	The COPIES operand on a SUBMIT.PRINT service request is not valid. This is a system error. Contact the IBM Support Center.
214	Service request	The data set on a SUBMIT.PRINT service request cannot be found. This is a system error. Contact the IBM Support Center.
215	File control/request	You attempted to store a record with a duplicate key while using an index that does not allow duplicate keys. The VisualAge Generator mnemonics UNQ and DUP are set.
216	File open/connect	A connection was attempted to an ESDS file or transient data in direct mode. The VisualAge Generator mnemonic FMT is set.
	Service request	The data set specified on a DEFDS request matches a CONNECT already in use. This is a system error. Contact the IBM Support Center.
217	File open/connect	An attempt was made to subconnect to a serial file. Check to see if a called program is attempting to reference the same serial file that has been referenced by the calling program.

Table 47. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
	Service request	<p>A PRINT error has occurred for one of the following reasons:</p> <ul style="list-style-type: none"> • An error occurred while writing to the transient data queue on MVS CICS. The most common errors are QIDERR, IOERR, LENGERR, and NOSPACE. • An error occurred while writing to the data definition name (DD name) EZEPRINT in MVS/TSO or MVS Batch. A possible cause is that the printer file (for example, EZEPRINT) has been allocated incorrectly or not been allocated at all. You get this error, for example, if the data set allocated for the print output has the wrong record format (anything other than VBA) or the wrong record length (shorter than the print output line length).
218	Service request	The file is not available. The VisualAge Generator mnemonic FNA is set.
22A	File control/request	The available storage space has been exhausted. Try the operation again. If the problem persists, contact your system programmer.
220	File open/connect	<p>A format error occurred. Either the characteristics of a file are not supported by VisualAge Generator Developer, or they are incompatible with the VisualAge Generator Developer record. The VisualAge Generator mnemonic FMT is set.</p> <p>In MVS/TSO, for example, this occurs if no member name was specified for gaining access to a PDS data set.</p>
	File control/request	The record length for a file is larger than the maximum record length defined in the system.
221	Service request	An ENQ was not successful while writing to the transient data queue on MVS CICS. This is a system error. Contact the IBM Support Center.

Table 47. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
223	Service request	The attach of the print subtask was not successful, or the print subtask abended. This is a system error. Contact the IBM Support Center.
225	Service request	The print subtask abended. This is a system error. Contact the IBM Support Center.
226	File control/request	An IO error occurred while reading or writing from temporary storage on MVS CICS. This is a system error. Contact the IBM Support Center.
25A	File control/request	The data set cannot be extended because VSAM cannot allocate additional direct-access storage space. Either not enough space is left to make the secondary allocation request, or you attempted to increase the size of a data set while processing with SHROPT=4 and DISP=SHR. The VisualAge Generator mnemonic FUL is set.
251	File open/connect	For MVS CICS or VSE CICS environments, the file control table (FCT) entry was not found, indicating a real file or transient data queue was not properly defined or generated. For MVS/TSO environments, either the DD name has not been allocated, or the data set for the dynamic allocation does not exist.
280	File control/request	The data set that you are trying to connect to is already in use. A probable cause is that your program has a data set associated to one record and you are trying to use another record with the same data set associated. You need to issue a CLOSE on the first record to free the data set before trying to use it with another record.
291- 294	Service request	A mapping error occurred.
	Terminal support	A mapping error occurred.

Table 47. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
380	File control/request	A deadlock occurred. One transaction is attempting to update a record that is currently locked by another transaction. The VisualAge Generator mnemonic DED is set.
381	File control/request	The control interval for a record is already held in exclusive control by another requester. The VisualAge Generator mnemonic LOC is set. For MVS CICS or VSE CICS systems, the returned code is INVREQ. This is assumed to have occurred due to one transaction's attempt to do two UPDATES to the same file. If this is not the case, see the description of INVREQ in the <i>CICS Application Programmer's Reference</i> manual.
389	File control/request	The resource control block could not be found to process the request against. This is a system error. Contact the IBM Support Center.
399	File control/request	You attempted to store a record to a temporary storage queue with a key that exceeds 32767. The key is too large for temporary storage queues, which cannot have more than 32767 records.
4nn	File open/connect	For MVS/TSO systems only, the VSAM GENCB for an ACB was not successful. The value of nn is determined from VSAM return codes. If register 15 contains 4, nn is replaced by the contents of register 0. If register 15 does not contain 4 (or 0), nn is replaced by 50 plus the contents of register 15.

Table 47. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
5nn	File open/connect	For MVS/TSO systems, an OPEN request is not successful. For VSAM files on MVS/TSO, a SHOWCB for the ERROR field is done after the problem with the OPEN request. The value of the ERROR field replaces nn . For non-VSAM sequential files in the MVS/TSO environment (QSAM), nn is replaced with a value of 0. For spool files in a CICS environment, nn is also replaced with 0.
5A0	File open/connect	An attempt was made to open a VSAM data set for input, but the data set was empty.
6nn	File open/connect	The VSAM GENCB for an RPL was not successful. For MVS/TSO, the value of nn is determined from VSAM return codes. If register 15 contains 4, nn is replaced by the contents of register 0. If register 15 does not contain 4 (or 0), nn is replaced by 50 plus the contents of register 15.
701	File open/connect	On MVS CICS 2.1 or later, the file cannot be opened or connected. The error is not defined in the FCT flags.
702	File open/connect	The VSAM SHOWCB or MODCB macro was not successful. This usually means that the file is not open.
703	File open/connect	The VSAM TESTCB macro was not successful.
705	File open/connect	For MVS CICS only, a connection was attempted to transient data or a temporary storage queue, but a VSAM file has the same name.
706	File open/connect	On MVS CICS 2.1 or later, the file is UNENABLED and cannot be opened or connected.
707	File open/connect	On MVS CICS 2.1 or later, the file is DISABLED or DISABLING and cannot be opened or connected.
708	File open/connect	On MVS CICS 2.1 or later, the user is not authorized to have access to the file.

Table 47. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
709	File open/connect	On MVS CICS 2.1 or later, an I/O error occurred on the SET data set Open command.
768	File open/connect	The OPEN or connection was not successful due to a GETMAIN error when requesting storage for control blocks associated with sequential files.
8nn	File control/request	These return codes do not have a VisualAge Generator mnemonic. Either MVS CICS returned an ILLOGIC error, or VSAM returned a return code of 8 on file input/output. The nn is replaced with the VSAM error code. For more information, see the <i>MVS VSAM administration Manual: Macro Instruction Reference</i> manual.
	File open/connect	A storage allocation was not successful.
80C	Storage allocation	There is insufficient storage to satisfy a GETMAIN request.
802	File open/connect	The resource is not associated.
	Storage allocation	There is insufficient storage for allocation.
803	Contents control	The module specified on a LOAD macro could not be located.
804	File control/request	This return code is received from MVS CICS or VSE CICS and indicates that a QIDERR or ITEMERR occurred while trying to gain access to a temporary storage queue.
805	Contents control	The module specified on a LOAD macro is damaged.
	Message processing	An unsupported option was specified on an INFORM macro.
806	Contents control	For MVS/TSO, the module specified on a LOAD macro could not be located.
807	Contents control	Insufficient storage is available to load the specified module.
81C	File Control/ Request	A Temporary Storage queue is full. The VisualAge Generator mnemonic FUL is set.

Table 47. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
987	File control/request	For MVS/TSO only, a branch was made to the SYNAD routine as the result of a GET to a non-VSAM serial file. A possible reason is that the file is empty or the blocking factor is wrong.
988	File control/request	For MVS/TSO only, a branch was made to the SYNAD routine as the result of a PUT to a non-VSAM serial file. A possible reason is that the file is empty or the blocking factor is wrong. For MVS CICS or VSE CICS, a WRITE request to a spool file was not successful.
989	File control/request	An error occurred while trying to close a file.
999	File control/request	An unsupported request was made for a serial file. A probable cause is that the VisualAge Generator record associated with this file was not defined with a serial organization.

Common SQL Return Codes

Only the most frequently occurring SQL codes are listed in this section. If you receive other SQL codes or if you need a more complete explanation of one of the SQL codes, refer to the documentation for your release of DB2.

RC Meaning

- 100** No rows were found by SQL that meet the search criteria specified in the WHERE clause of the SQL statement, or if processing a SCAN process option in conjunction with a SETINQ or SETUPD process option, the end of the selected rows has been reached. The possible causes are the following:
- The key value(s) were not moved correctly to the host variable(s) used in the WHERE clause.
 - No rows meet the search criteria specified in the WHERE clause.
 - VisualAge Generator Developer stripped trailing blanks for the character host variables used in a LIKE predicate in the WHERE clause. You can use the /TRACE=SQLIO generation option to enable tracing of the data sent to SQL and the data coming back

from SQL. See “VisualAge Generator Server Trace Facility” on page 230 for more information about using the VisualAge Generator Server trace facility.

-301, -302, -303, -304

The VisualAge Generator Developer data item definition does not match the definition of the same column in the DB2 table. This can be caused by defining a column as variable length, but not defining the data item in VisualAge Generator Developer with a variable-length SQL code. This can also be caused by specifying a different length to VisualAge Generator Developer from what was defined in the DB2 table.

Make the necessary changes in the VisualAge Generator Developer data item definition to match the DB2 table and generate the program again.

-302 For the IMS/VS environment, you might have allocated the DB2 work database with a 4KB page size instead of the required 32KB page size.

Refer to the program directory in the appendix of this manual for information about installing a DB2 work database.

-805 The DBRM for the current program was not bound as part of the current DB2 plan. Possible causes are:

- The BIND process was never run for the program.
- An incorrect plan name was specified at startup.
- The plan name specified in the RCT on MVS CICS did not match the plan name used in the BIND process.
- All programs that run together under a single transaction or job step must be bound into the same DB2 plan.

Look at the message inserts to see what DB2 returned as the program name and plan name. If these are what you expect, review the steps used for preparing the program.

-818 The DB2 precompiler-generated time stamp in the load module is different from the database request module (DBRM) used on the most recent BIND for the PLAN being used. Both of these precompiler outputs must match and one of them is not from the most recently-run precompile. This typically happens when the precompile, link-edit, and bind process is run more than once and either the DBRM library or the load library used for the load module is changed. This creates the opportunity to pick up the old load module at run time if the old load library is first in the search sequence at run time. Alternatively, the BIND process might be using an old DBRM library that contains an old copy of that member.

Another possible cause when running on MVS CICS or IMS/VS is that an old copy of the load module is resident in storage.

Ensure that you are running with the most recent copy of the load module and that you are using the same DBRM library on the precompile and BIND steps. On MVS CICS ensure that the latest copy of the load module has been picked up by issuing an MVS CICS NEWCOPY command or by using the VisualAge Generator Server new copy utility. On IMS/VS ensure that the latest copy of the load module has been picked up by recycling the message region.

-911,-913

A deadlock condition occurred. Possible causes are:

- The isolation level was set for repeatable read.
- There were long periods of time between commit points.
- The program issued an update process without a replace option.

Note: The program should be coded to handle these conditions.

-922

Connection authorization was not successful. The type of error is indicated in the SQL error message. Some typical causes are not granting authority for the DB2 plan or not creating a synonym for one or more of the DB2 tables.

Make the necessary changes to provide authorization to the DB2 plan and then run the program again. You might also want to refer to the documentation for your release of DB2 for additional causes of the authorization error.

Common DL/I Status Codes

Only the most frequently occurring DL/I status codes are listed in this section. If you receive other DL/I status codes or if you need a more complete explanation of one of the DL/I status codes, refer to the application programming manual for your release of IMS.

Code Meaning

AD The function parameter on the call is not valid. If the function code is correct, the status code can be from an I/O or alternate PCB for a database call. You might have a mismatch between the PSB you defined in VisualAge Generator Developer and the IMS PSB definition.

AI A data management open error occurred. Either no DD statements were supplied for logically related databases, or the DD name is not

the same as the name specified on the DATASET statement of the DBD. The segment name area in the DB PCB has the DD name of the data set that could not be opened.

- AJ** The format of one of your SSAs is not valid. Either the SSA contains a command code for that call that is not valid, or the SSA uses an R, S, W, or M command code for a segment for which there are no subset pointers defined in the DBD.
- AK** An SSA contains either a field name that is not valid or a name that is not defined in the DBD, or the VisualAge Generator Developer data item for DL/I segment does not match the name defined to DL/I.
- AM** The call function is not compatible with the processing option in the PCB, the segment sensitivity, the transaction-code definition, or the program type.
- GA** In trying to satisfy an unqualified GN or GNP call, DL/I crossed a hierarchic boundary into a higher level.
- GB** In trying to satisfy a GN, DL/I reached the end of the database.
- GD** The program issued an ISRT that was not qualified for all levels above the level of the segment being inserted. The segment might have been deleted by a DLET using a different DB PCB.
- GE** DL/I is unable to find a segment that satisfies the segment described in a get call.
- GK** DL/I has returned a different segment type at the same hierarchic level for an unqualified GN or GNP.
- GP** The program issued a GNP when parentage is not established, or the segment level specified in the GNP is not lower than the level of the established parent.
- II** The program issued an ISRT that tried to insert a segment that already exists in the database.

Common VSAM Status Codes

Only the most frequently occurring VSAM codes are listed in this section. If you receive other VSAM codes or if you need a more complete explanation of one of these values, refer to the *MVS VSAM Administration: Macro Instruction Reference* manual.

OPEN request type

Code	Meaning
------	---------

- | | |
|-----------|--|
| 64 | Warning message: OPEN encountered an empty alternate index that is part of an upgrade set. |
|-----------|--|

- 74** This is a warning message indicating the data set was not properly closed. Either the implicit verify for the OPEN was unsuccessful, or the user specified that the implicit verify should not be attempted for the OPEN. A previous VSAM program might have ended abnormally. The VERIFY command of Access Method Services can be used to properly close the data set.
- 80** The DD statement for this access method control block is either missing or not valid.
- 94** Either no record for the data set to be opened was found in the available catalog or catalogs, or an unidentified error occurred while VSAM was searching the catalog.
- 98** Security verification was not successful; the password specified in the access method control block for a specified level of access does not match the password in the catalog for that level of access.
- A0** The operands specified in the ACB or GENCB macro are inconsistent either with each other or with the information in the catalog record. You might have attempted to open an empty data set for input only (SCAN).
- A8** The data set was not available for the type of processing you specified, or an attempt was made to open a reusable data set with the reset option while another user had the data set open.
- BC** The data set indicated by the access method control block is not a valid type of data set for specification by an access method control block. You might have used a sequential data set as the physical file, but specified VSAM or VSAMRS as the file type for resource association when you generated the program.
- C0** An unusable data set was opened for output.
- C4** Access to data was requested using an empty path.

CLOSE request type

- | Code | Meaning |
|-------------|--|
| 04 | The data set indicated by the access method control block is already closed. |
| 88 | Not enough virtual storage was available in the address space of your program for the work area required by CLOSE. |
| 94 | An unidentified error occurred while VSAM was searching the catalog. |

GET/PUT/POINT/ERASE/CHECK/ENDREQ request types

Note: The following occur when register 15=8(8).

Code	Meaning
08	An attempt is made to store a record with a duplicate key, or there is a duplicate record for an alternate index with the unique key option.
6C	The RECLLEN specified was one of the following: <ul style="list-style-type: none"> • Larger than the maximum allowed • Equal to 0 • Smaller than the sum of the length and the displacement of the key field • Not equal to the record(slot) size specified for a relative record data set
70	The KEYLEN specified was too large or equal to 0.
C0	A relative record number that is not valid was encountered.

COBOL Status Key Values

This shows the most frequently occurring COBOL status key values. If you receive other status key values or if you need a more complete explanation for one of these values, refer to the application programming language reference for your release of COBOL.

Status Key	Explanation
10	The end of a file was reached.
22	An attempt was made to write a record with a key that duplicated one that was already in the file.
23	Record not found. This can also be caused by an optional file not being allocated.
35	No DD statement was included in the JCL. This can occur if the program calls another program or transfers to another program using a DXFR statement, but the DD statements for the second program have not been added to the sample runtime JCL for the main program.
39	The physical file that you specified during resource association does not match the file characteristics that you specified during record definition. The file characteristics include file organization (sequential, relative or indexed), the prime record key, the alternate record keys, and the maximum record size.
44	A variable-length record was written that is not valid. This can occur if the value in the number of occurrences field is

larger than the maximum value, or the value in the record length item field is larger than the maximum length of the record.

96

No DD statement was included in the JCL for a VSAM file. This can occur if the program calls another program or transfers to another program using a DXFR statement, but the DD statements for the second program have not been added to the sample runtime JCL for the main program.

Chapter 28. VisualAge Generator Server Return Codes and Abend Codes for MVS Systems

The information within this chapter is diagnosis, modification, or tuning information.

Only the most frequently occurring IMS and system abend codes are listed in this section. If you receive other abend codes or if you need a more complete explanation of one of the codes, refer to the IMS messages and codes manual for your release of IMS.

Return Codes

This section contains a listing of codes set by VisualAge Generator Server and returned in the COBOL return code of a program.

If a generated program completes normally, the COBOL return code is set to the value in the EZERCODE special function word. This code must be less than or equal to 512. Return codes greater than 512 are reserved for VisualAge Generator Server. The return codes set by VisualAge Generator Server are:

- 693** The program ended due to an error detected by VisualAge Generator Server. The error description is reported as described in Chapter 25. Diagnosing Problems for VisualAge Generator Server on MVS Systems.
- 4093** A program generated using VisualAge Generator Server Version 1.2 ended due to an error detected by VisualAge Generator Server.

If LE detects an error and returns to the operating system, the LE return code modifier (2000 - error, 3000 - severe error, or 4000 - critical error) is added to the user or VisualAge Generator Server return code.

ABEND Codes

VisualAge Generator Server reports errors by error messages whenever possible. ABENDs are issued only in situations where initialization has not progressed to the point where messages can be issued or when the error messages cannot be written to their normal destination.

MVS CICS Environments

For MVS CICS, you can control whether or not a dump is taken by using the diagnostic controller utility. If a dump is taken, the dump code is ELAD. See “Controlling Error Reporting in MVS CICS” on page 214 for information on the diagnostic controller utility.

ELA1 This abend code should never be received. However, if register 1 in a dump contains “ELA1”, then a database manager or subsystem interface module (for example, ASMTDLI for DL/I access) was not linked with a VisualAge Generator Server program at product installation. Registers 3 and 4 in the dump usually contain the name of the stub program. The abending load module is the module that was not linked correctly.

ELA2 The Task Work Area (TWA) does not exist or is not long enough to be used by VisualAge Generator Server. The TWA length must be greater than or equal to the sum of 1024 plus the /TWAOFF (TWA offset) generation option specified when the initial program in the transaction was generated.

Use the TWASIZE parameter in the PCT entry to define a TWA with an adequate length for the transaction.

When this is a client/server program set, ensure that the CICS mirror transactions on the server CICS have a TWA size at least equal to the TWA size specified for the VisualAge Generator transaction on the client system.

ELA3 Load for module ELARSCNT was not successful. VisualAge Generator Server has not been installed correctly.

Ensure the MVS CICS region can gain access to the VisualAge Generator Server runtime library and that module ELARSCNT is defined in the PPT.

When this is a client/server program set, ensure that the security level-checking parameters (RSL and RSLC) for the CICS mirror transactions on the server CICS are the same as those specified for the VisualAge Generator transaction on the client system.

ELA4 Load for module ELARPRTRX was not successful. VisualAge Generator Server has not been installed correctly.

Ensure the MVS CICS region can gain access to the VisualAge Generator Server runtime library and that module ELARPRTRX is defined in the PPT.

ELA5 Load for module ELARPRTC was not successful. VisualAge Generator Server has not been installed correctly.

Ensure the MVS CICS region can gain access to the VisualAge Generator Server runtime library and that module ELARPRTC is defined in the PPT.

ELA6 The dynamic storage stack used for working storage for VisualAge Generator Server modules was exhausted and VisualAge Generator Server could not continue.

This problem should not occur. Report the problem to the IBM support center.

ELA7 A GETMAIN was not successful. There was not enough storage for the program to complete.

Try the program again when the region is less busy or try it again in a larger region.

ELA9 Load or link for a VisualAge Generator Server module was not successful. VisualAge Generator Server has not been installed correctly. Use CEDF to determine the module name. Look for a PGMIDERR on a CICS LOAD or CICS LINK command.

Ensure that the MVS CICS region can gain access to the VisualAge Generator Server runtime library and the module name being loaded is defined in the PPT.

ELAB A call was made to a main program, which is not allowed, or a non-VisualAge Generator program was transferred to with a DXFR statement and the NONCSP parameter was not specified either on the DXFR statement or as the LINKTYPE parameter in the linkage table.

ELAC VisualAge Generator Server has detected a FREEMAIN request that is not valid. Collect the dump and contact the IBM Support Center for assistance.

ELAE A generated program has ended because of a serious error. This occurs for one of the following reasons:

- Storage has been corrupted so that a dump is necessary to debug the abend.
- Error handling was unable to write messages to the error destination queue or to the user at the terminal. The dump is necessary to make the diagnostic information available. The situation can occur if the error destination queue specified for the transaction using the diagnostic controller utility is not defined to MVS CICS. In MVS CICS, if the error destination queue is defined as an intrapartition queue, this situation occurs when there is no more space on the intrapartition queue and the error messages cannot be written.

- A severe error has occurred. Refer to the error destination queue for the corresponding error messages. The default name is ELAD. The queue name can be changed using the diagnostic controller utility.

See “VisualAge Generator Server ABEND Dumps” on page 227 for information on how to find error messages in the dump on an ELAE abend.

ELAF ELATSRST has detected one of the following errors:

- ELATSRST was not initiated with a CICS XCTL command (for example, the restart transaction ID was associated directly to ELATSRST).
- The COMMAREA length on entry was not 0 or 10.
- The VisualAge Generator Server portion of the TWA had been initialized, indicating that a CONVERSE process option was not in process or the non-VisualAge Generator program uses the TWA and the program was not generated with the proper TWA offset.
- Information in the COMMAREA was not valid indicating that a CONVERSE process option was not in process.
- Information in the COMMAREA indicates that this module was started with an XFER statement with a map or during First Map processing.

ELAW A program and its associated map groups or a map group and its associated tables were generated using incompatible versions of COBOL generators. For more information, refer to the section on transferring program control in the *VisualAge Generator Design Guide*. In addition, use the search function to locate APAR PN45782 information.

IMS, IMS BMP, MVS/TSO, and MVS Batch Environments

- 1600** A generated program has ended because of a serious error. This occurs for one of the following reasons:
- Storage has been corrupted so that a dump is necessary to debug the abend.
 - Error handling was unable to write messages to the error destination queue or to the user at the terminal. The dump is necessary to make the diagnostic information available. In IMS, the situation can occur if the error destination queue specified using the /ERRDEST generation option is not defined to IMS.
 - A severe error has occurred. In IMS, refer to the error destination queue specified using the /ERRDEST generation option for the corresponding error messages. In MVS/TSO and MVS Batch, refer to the data set ELAPRINT for the messages.

See “VisualAge Generator Server ABEND Dumps” on page 227 for information on how to find error messages in the dump on a 1600 abend.

- 1601** A database manager or subsystem interface module (for example, ASMTDLI for DL/I access) was not linked with a VisualAge Generator Server program at product installation. Registers 3 and 4 in the dump contain the name of the stub program. The abending load module is the module that was not linked correctly.

Refer to the *Program Directory for VisualAge Generator Server for MVS* for information on correctly linking the abending load module.

- 1602** A program generated with the /FASTPATH option ended because of a rununit error. The abend is issued to prevent any further scheduling of the program in error.

See “VisualAge Generator Server ABEND Dumps” on page 227 for information on how to find error messages in the dump on a 1602 abend. Depending on the generation options specified for the program, the message might also have been written to an error diagnostic message queue, on the IMS log, or to an ELAPRINT file. See Chapter 25. Diagnosing Problems for VisualAge Generator Server on MVS Systems for more information on VisualAge Generator Server error reporting.

- 1606** The dynamic storage stack used for working storage for VisualAge Generator Server modules was exhausted and VisualAge Generator Server could not continue.

This problem should not occur. Report the problem to the IBM Support Center.

- 1608** VisualAge Generator Server has detected a FREEMAIN request that is not valid. Collect the dump and contact the IBM Support Center for assistance.

Chapter 29. Codes from Other Products for MVS Systems

The chapter contains lists of common system abend codes, COBOL runtime messages, LE abend codes, common IMS runtime messages and abend codes, and common runtime messages from MVS CICS.

Common System Abend Codes for All Environments

Only the most frequently occurring abend codes are listed in this section. If you receive another abend code or if you need a more complete explanation of one of the abend codes, refer to the *System Codes* manual for your release of MVS.

System 0C4 This code can occur on a DXFR statement if there is a print services or table program with the same name as the transferred-to program. This code can also occur when a print services or table program is called but there is a different program (for example, a non-VisualAge Generator program or an application program) with the same name. Using naming conventions can eliminate this problem.

This code can also occur if you add table edits to a map group that is shared by multiple programs and do not generate all the programs again.

System 0C7 Data exception. The abend occurs when fields defined as decimal or packed decimal are retrieved from a database and are found to contain data of a different format.

The abend can also occur if fields that are not initialized are used in calculations or comparisons. This happens if the program attempts to read a record from a database and the record is not found, but the program uses fields in the record anyway. To ensure that records are initialized, use a SET record EMPTY statement in the program or specify /INITADDWS and /INITRECD as generation options. Refer to the VisualAge Generator online help system for additional information on how to initialize records using a SET record EMPTY statement in the program. Scan the helps using the phrase SET record EMPTY.

The abend can also occur when SET record EMPTY is used or when /INITADDWS and /INITRECD are used if one of the following is true:

- There are redefined records with different data types or variable field boundary alignments from the original record.
- The primary working storage record receives a transferred record that contains different data types or variable-field boundary alignments from the original record.

System 806

Module not found in a library. This can occur if a new version of a module is put into a load library and is placed in secondary extents. To avoid this when you allocate load libraries, specify a large primary allocation and 0 for the secondary allocation. This insures that if there is enough space for the load module it will be placed in the primary extent. If there is not enough space, there will be an abend (for example, a B37 abend for insufficient space) when you link the module into the load library. Using this technique detects the space problem during the preparation step rather than at run time.

In IMS, this can occur if a program transfers to another program using an XFER statement and the transaction named on the XFER statement is defined in the IMS system definition, but the load module for the program is not in a library available to the IMS message region.

In other environments, this can occur if the module is not in a library defined in your link list, JOBLIB, or STEPLIB concatenation sequence.

If the missing module name is ELACxxx, the NLS language code identified by the last 3 characters of the module name is not installed on the system. This language code was specified with the /NLS generation option when the program was generated.

See “Languages Available” on page 587 for a list of valid NLS language codes.

If the module that could not be loaded is ELARSCNT and you are running in the IMS/VS, IMS BMP or MVS batch environment, check that you have VisualAge Generator Server installed. If you generate a program using &370ad; or VisualAge Generator Developer V1R1 and then attempt to run it using VisualAge Generator Server, you will get an 806 when ELARSCNT cannot be loaded.

LE Runtime Messages

Only the most frequently occurring LE runtime messages are listed in this section. If you receive other runtime messages that start with IGZ or if you need a more complete explanation of one of the messages, refer to the debugging manual for your release of LE.

IGZ0033S **An attempt was made to pass a parameter address above 16 megabytes to AMODE(24) program program-name.**

Explanation: An attempt was made to pass a parameter located above the 16-megabyte storage line to a program in AMODE(24). The called program will not be able to address the parameter.

Programmer Response: If the calling program is compiled with the RENT option, the DATA(24) option may be used in the calling program to make sure that its data is located in storage accessible to an AMODE(24) program. If the calling program is compiled with the NORENT option, the RMODE(24) option may be used in the calling program to make sure that its data is located in storage accessible to an AMODE(24) program. Verify that no linkedit, binder or genmod overrides are responsible for this error.

System Action: The application was terminated

IGZ0064S **A recursive call to active program program-name in compilation unit compilation-unit was attempted.**

Explanation: COBOL does not allow reinvocation of an internal program which has begun execution, but has not yet terminated. For example, if internal programs A and B are siblings of a containing program, and A calls B and B calls A, this message will be issued.

Programmer Response: Examine your program to eliminate calls to active internal programs.

System Action: The application was terminated.

IGZ0066S **The length of external data record data-record in program program-name did not match the existing length of the record.**

Explanation: While processing External data records during program initialization, it was determined that an External data record was previously defined in another program in the run unit, and the length of the record as specified in the current program was not the same as the previously defined length.

Programmer Response: Examine the current file and ensure the External data records are specified correctly.

System Action: The application was terminated.

IGZ0075S **Inconsistencies were found in EXTERNAL file file-name in program program-name. The following file attributes did not match those of the established external file: attribute-1 attribute-2 attribute-3 attribute-4 attribute-5 attribute-6 attribute-7**

Explanation: One or more attributes of an external file did not match between two programs that defined it.

Programmer Response: Correct the external file. For a summary of file attributes which must match between definitions of the same external file, see *IBM COBOL Language Reference*.

System Action: The application was terminated.

COBOL Runtime Messages

Only the most frequently occurring COBOL runtime messages are listed in this section. If you receive other runtime messages that start with IGZ or if you need a more complete explanation of one of the messages, refer to the *Debugging and Runtime Messages Guide* for your release of LE.

IGZ033S **An attempt was made to pass a parameter address above 16 megabytes to AMODE(24) program *program-name*.**

Explanation: An attempt was made to pass a parameter above the 16-megabyte storage line to a program in AMODE(24). The called program will not be able to address the parameter.

IGZ064S **A recursive call to active program *program-name* in compilation unit *compilation-unit* was attempted.**

Explanation: COBOL does not allow an internal program that has started to run, but has not completed, to be invoked again. For example, if internal programs A and B are siblings of a containing program, and A calls B and B calls A, this message will be issued.

IGZ066S **The length of *external data record data-record* in program *program-name* did not match the existing length of the record.**

Explanation: While processing external data records during program initialization, it was

determined that an external data record was previously defined in another program in the run unit, and the length of the record as specified in the current program was not the same as the previously defined length.

IGZ075S **Inconsistencies were found in EXTERNAL file *file-name* in program *program-name*. The following file attributes did not match those of the established external file: *attribute-1 attribute-2 attribute-3 attribute-4 attribute-5 attribute-6 attribute-7***

Explanation: One or more attributes of an external file did not match between two programs that defined it.

Common COBOL Abend Codes

Only the most frequently occurring abend codes are listed in this section. If you receive another abend code or if you need a more complete explanation of one of the messages, refer to the debugging manual for your release of LE.

User 4087 This is an LE abend code. If reason code is 7, the error could be due to the region size not being large enough to run the COBOL program.

Common IMS Runtime Messages

Only the most frequently occurring IMS runtime messages are listed in this section. If you receive another runtime message that starts with DFS or if you need a more complete explanation of one of the messages, refer to the IMS messages and codes manual for your release of IMS.

DFS057I **REQUESTED BLOCK NOT AVAILABLE: blockname RC = reason code**

Explanation: The blockname is either the MOD or the DOF name. If it is the DOF name, the first 2 bytes of the name are the device type and features printed in hexadecimal. Refer to the message format services manual for your release of IMS for an interpretation of these 2 bytes. If it is a MOD name, it will be the name of a map group.

User Response: If a DOF name was specified, review the values you specified for the /MFSDEV, /MFSEATTR, and /MFSIGNORE generation options, and compare them to the IMS system definition for the terminal that had the problem.

If a MOD name was specified, ensure that you installed the MFS control blocks into the correct library. If you specified the /MFSTEST generation option, ensure that you used the /TEST MFS command. If you specified /NOMFSTEST, ensure that your system administrator has run the IMS online change utility to copy in the new format definitions.

DFS064 **NO SUCH TRANSACTION CODE**

Explanation: This message is sent to a terminal when the transaction code requested by the user is not defined to IMS. An example of a situation that results in this message is when a program does an XFER with a map to a transaction that is not defined to IMS. The map is written to the terminal, but when the user enters data, the transferred-to transaction cannot be scheduled because it is not defined to IMS.

User Response: Either ensure the transaction code is defined to IMS or change the XFER

statement in the transferring program to reference the correct IMS transaction code.

DFS182 **INVALID OR MISSING PARAMETER**

Explanation: An IMS reserved word (for example, LTERM) was used as a map name in a /FORMAT command.

User Response: If you need to use the /FORMAT command to display this map, you need to change the map name and generate the map group and any programs that use this map again.

DFS555I **TRAN tttttttt ABEND S000,Uaaaa; MSG IN PROCESS: (up to 78 bytes of data) time stamp**

Explanation: This message indicates that the transaction running in IMS has ended abnormally. Typical abend codes are shown below:

0778 IMS user abend, indicating that a ROLL request was issued

1602 VisualAge Generator Server abend because a rununit error occurred in a program that was generated with the /FASTPATH option

1600 VisualAge Generator Server abend because an unrecoverable error occurred in situations other than rununit errors for /FASTPATH programs

User Response: Press the PA1 or PA2 key to display the error map that contains the error diagnostics that describe the error.

**DFS2082 RESPONSE MODE TRAN
TERMINATED WITHOUT REPLY**

Explanation: VisualAge Generator Server has ended the logical unit of work for a program that was generated with the /FASTPATH option.

User Response: Press the PA1 key to display the error map that contains the error diagnostics that describe the error.

DFS2766I PROCESS FAILED

Explanation: IMS issues this message if VisualAge Generator Server ends the run unit for a transaction program that was generated with /FASTPATH and run in an IMS fast-path region.

User Response: Press the PA1 or PA2 key to display the error map that contains error diagnostics that describe the error. See Chapter 25. Diagnosing Problems for VisualAge Generator Server on MVS Systems for additional information.

(none) Logged off IMS and returned to the VTAM sign-on screen without any warning or error message being displayed.

Explanation: One of the following might have occurred:

- The program attempted to display a map with DBCS or mixed data on a non-DBCS terminal or printer.
- The values specified for the /MFSDEV, /MFSEATTR, and /MFSIGNORE generation options do not match the IMS system definition for the terminal that had the problem.

User Response: Correct the program or generation options, generate the program and map group again, and then run the program again.

Common IMS Runtime Abend Codes

Only the most frequently occurring IMS abend codes are listed in this section. If you receive another abend code or if you need a more complete explanation of one of the abend codes, refer to the messages and codes manual for your release of IMS.

- IMS 259** A program has been compiled with the DATA(31) compile option and is being run in a non-IMS/ESA environment. The program should be recompiled with the DATA(24) compile option.
- IMS 462** A program was scheduled in a message region, but the program ended without successfully issuing a get unique for an input message. This can occur if VisualAge Generator Server detects an error that would prevent the program from processing properly. Examples of these errors are:
- The IMS PSB does not match the PSB defined in VisualAge Generator Developer.
 - The print services program is missing.
- IMS 778** A ROLL call has been issued VisualAge Generator Server because of a rununit error or a catastrophic error in the IMS/VS environment. The ROLL is issued to prevent further scheduling of the program in error. IMS displays message

DFS555I indicating that abend 778 has occurred. The VisualAge Generator Server error message panel can be displayed by pressing PA1.

See Table 28 on page 208 for the types of errors that might cause a rununit error. Based on your generation options and the JCL for your message region, additional diagnostic information might be provided on an error diagnostic message queue, on the IMS log, or in ELAPRINT. See “Controlling Error Reporting in IMS Environments” on page 214 and the appropriate ELA messages in Chapter 61. VisualAge Generator Server Messages for MVS, VSE, and VM Systems for additional information.

Note: Press PA2 if PA1 does not cause the VisualAge Generator Server error map to display.

IMS 1008 A program that was running as a BMP and that obtained access to fast-path databases did not issue a SYNC or CHKP call at the end of the job step. You can force the CHKP call to occur by:

- Using the EZECOMIT special function word in a batch-oriented BMP
- Ensuring that the transaction-oriented BMP ends with an EOF (QC status) for the file being used for input from the IMS message queue

IMS 3042 Access to DB2 cannot be obtained. Possible causes of this are:

- The terminal ID is not defined to DB2.
- The DB2 program plan is not valid or access to the DB2 program plan cannot be obtained.

If the program was being run as a BMP, see Figure 29 on page 176 for sample JCL.

Common MVS CICS Runtime Messages

Only the most frequently occurring MVS CICS runtime messages are listed in this section. If you receive another MVS CICS runtime message that starts with DFH or if you need a more complete explanation of one of the messages, refer to the MVS CICS messages and codes manual for your release of MVS CICS.

DFHAC2016 date time applid **Transaction**
tranid **cannot run because**
program program-name **is not**
available.

Explanation: The transaction tranid cannot be run because the initial program for the transaction is not available. This could occur because the transaction is defined in the PCT, but the program is not defined in the PPT or is not in a library in the DFHRPL concatenation.

User Response: Have your system administrator check the PPT entries. Be sure the program is in a library in the DFHRPL concatenation.

DFHAC2206 time applid **Transaction** tranid **has**
failed with abend abcode.
Resource backout was successful.

Explanation: The transaction tranid has ended abnormally with abend code abcode. abcode is either an MVS CICS transaction abend code or a user abend code.

User Response: If the user abend code starts with ELA, see “MVS CICS Environments” on page 274 . If it is an MVS CICS abend code, see “Common MVS CICS Abend Codes” to see if it is included there. If not, refer to the MVS CICS messages and codes manual for your release of MVS CICS.

Common MVS CICS Abend Codes

Only the most frequently occurring MVS CICS abend codes are listed in this section. If you receive another MVS CICS abend or if you need a more complete explanation of one of the abend codes, refer to the MVS CICS messages and codes manual for your release of MVS CICS.

Depending on your diagnostic options, information might be available on an error destination queue or in an MVS CICS journal. See “Controlling Error Reporting in MVS CICS” on page 214 and the appropriate ELA messages in Chapter 2.

- | | |
|-------------|--|
| ACN1 | The DFHCNV conversion table cannot be loaded. If the program running under CICS OS/2 attempts a DPL to a generated COBOL program running under MVS CICS, you need to provide the CICS conversion table, DFHCNV. Although the table is not used by VisualAge Generator Server, it is required by CICS/MVS. |
| ADCA | An error occurred while processing a DL/I request. In addition to looking for the information provided by MVS CICS, look for messages or abends from DL/I. |
| ADLD | A program isolation deadlock occurred and a transaction was selected for an abend. See “Transaction Entries (PCT)” on page 145 for information about making your programs restartable. Refer to the <i>VisualAge Generator Design Guide</i> for information on using the EZEDLRST special function word and for information on designing restartable transactions. |
| AEY9 | Access to DB2 cannot be obtained. This occurs if DB2 is not running. |

AFCY	A transaction was purged when a deadlock occurred because a file is defined with LSRPOOLID not equal to NONE in the FCT, and one process within a program has performed a SCAN against a file and another process requested an update or add to the same file (or its alternate index) without ending the SCAN. Change the LSRPOOLID to NONE, or change the program design to end the SCAN before the update or add is requested.
APCT	A requested module cannot be located in the PPT or in the program library.
ASRA	A program check occurred. Some of the reasons this can occur for a VisualAge Generator program are as follows: <ul style="list-style-type: none"> • Incorrectly linked VisualAge Generator Server modules. If register 1 contains ELA1, see the information for ELA1 in “MVS CICS Environments” on page 274. • Data not initialized or data initialized to incorrect values. If the error occurred as a result of a data exception, see the explanation for “System 0C7” in “Common System Abend Codes for All Environments” on page 279.
ATDD	The program attempted to process a transient data queue that is disabled. This can occur for a program file associated with a transient data queue or for the transient data queue used for error diagnostic information.
AXFQ	The most common cause is the result of INBFMH not being specified equal to ALL in the profile associated with the CICS mirror transaction (CPMI). If the program running under CICS OS/2 attempts a DPL to a generated COBOL program running under MVS CICS, you need to ensure that the profile associated with the CPMI transaction is defined with INBFMH=ALL.

Note: CICS users that receive abend codes ADLD, ADCP, AKCT, or D106 might see four question marks in place of the CICS abend code for the resulting VisualAge Generator Server message. The CSMT console log contains the true CICS abend code that was issued.

COBOL Abends under CICS

- 1009** A program has a dynamic storage requirement greater than 64KB, but was compiled with the DATA(24) compiler option. Compile the module again with the DATA(31) compiler option.

1029 Either a PPT entry for a program attached through a COBOL dynamic call is not found or the module being invoked cannot be found in the MVS CICS region program library search string. Additional information can be retrieved by entering transaction CEBR on the terminal where the error occurred.

Part 6. Administering on VSE Systems

Chapter 30. General System

Considerations for VSE Systems	291
Considerations that Affect Performance	291
Generation and Compiler Options	291
Phases in Memory	292
Files and Databases	292
Defining and Loading VSAM Program Data Files	292
Defining VSAM Data Sets	293
Defining an Alternate Index	295
Loading Data in the Files	296
Support for DBCS Terminals	297
Library Space Considerations	298
Extended Addressing Considerations for VisualAge Generator Server	298
Database Considerations	300
Preparing Programs	300
Checking Access Authorization	301
Moving Program Packages among VSE Systems	301
Fold Tables	302
Backing Up Data	302
Customizing VisualAge Generator Server	302

Chapter 31. System Considerations for

VSE CICS	303
Required File Descriptions	303
Segmented and Nonsegmented Processing	304
Using Transient Data Queues for Printing in VSE CICS	305
Processing a DISPLAY Process Option for a Printer Map	305
Specifying Print Destination	306
CICS Terminal Printing	306
Special Parameter Group for the FZETPRT Program	307
PRTBUF Parameter	308
PRTMPP Parameter	309
PRTTYTYP Parameter	309
FORMFD Parameter	310
CICS Entries for FZETPRT (DBCS only)	310
Using the New Copy Function	311
Recovery and Database Integrity Considerations	311

Specifying Recovery Options in the CICS Tables	312
Determining When Database Changes Are Committed	312
Considerations that Affect Performance	313
Residency (Phases in Memory) Considerations	313
Virtual Storage Considerations and Residency	314
Pseudoconversational Programs and Residency	315
Work Database Temporary Storage Queue Considerations	315
Conversational and Pseudoconversational Processing	316
Terminal Printing	316
Monitoring and Tuning the System	317
Using and Allocating Data Files in MVS CICS and VSE CICS	317
Defining and Loading VSAM Data Files	317
Adding the Job Control Statements	317
Adding the File Name to the CICS File Control Table	317
Using Remote Files	319
Defining Transient Data Queues	320
Defining Intrapartition Transient Data	320
Defining Extrapartition Transient Data	321
Considerations for Using DB2/VSE in VSE CICS	322
Connection Considerations	322
Recovery and Database Integrity Considerations	323
Considerations for Using DL/I in VSE CICS	323
DL/I Database Support	323
Recovery and Database Integrity Considerations	324

Chapter 32. System Considerations for

VSE Batch	325
Using and Allocating VSAM Data Files in VSE Batch	325
Using Sequential Data Files in VSE Batch	325
Determining When Database Changes are Committed	327

Considerations for Using DB2/VSE in VSE		Considerations for Using DL/I in VSE	
Batch.	328	Batch.	330
Connection Considerations	328	DL/I Support in VSE Batch Programs	330
Considerations for Processing DB2/VSE		Recovery and Database Integrity	
Programs	330	Considerations	331
Recovery and Database Integrity		Performance Considerations for VSE Batch	331
Considerations	330	Runtime JCL	331

Chapter 30. General System Considerations for VSE Systems

This chapter describes the system requirements and considerations for administering VisualAge Generator Server in all of the supported MVS environments.

The following information is discussed:

- Considerations that affect performance
- Defining and loading VSAM program data files
- Support for DBCS terminals
- Extended addressing considerations for VisualAge Generator Server
- DB2* considerations
- Fold tables
- Backing up data
- Customizing VisualAge Generator Server

Considerations that Affect Performance

Specifying certain generation and compiler options and making reentrant programs resident in memory can affect the performance of VisualAge Generator programs.

Generation and Compiler Options

Specifying the following generation options might improve runtime performance:

- /INEDIT=INONLY
- /MATH=COBOL
- /NODEBUGTRACE
- /NOINITADDWS
- /NOINITRECD
- /NOLEFTJUST
- /NONULLFILL
- /NONUMOVFL
- /NOSETFULL
- /NOVALIDMIX

Specifying the following compiler options might improve runtime performance:

- NOFDUMP
- NOSSRANGE
- NOTEST
- OPT

Note: Refer to the *VisualAge Generator Generation Guide* for details on these options.

Phases in Memory

Placing phases in memory can improve performance by reducing the number of I/O operations (EXCPs). Phases can be placed in memory by loading them in the shared virtual area (SVA) or by defining them as resident to VSE CICS. Doing this can improve performance by preventing multiple loads of the program. See the *Program Directory for VisualAge Generator Server for VSE* (GI10-0813) for information about the modules that are reentrant (SVA-eligible).

Generated programs, print service programs, map group format modules, and shared tables (and single user tables in CICS) are also reentrant and can be included in the SVA or defined as resident to CICS. If a table is modified in CICS, a copy of the original table is saved to read/write storage before the modification is made.

Files and Databases

Standard tuning techniques (such as buffering) can be used with files and databases used by generated COBOL programs.

Defining and Loading VSAM Program Data Files

This section describes how to define and load VSAM data sets for use as program data files in the MVS CICS, MVS/TSO, or MVS batch environments. The section contains the following information:

- Defining VSAM data sets
- Defining an alternate index
- Loading data into the files

Defining VSAM Data Sets

VSAM data files can be serial (ESDS), relative (RRDS), or indexed (KSDS) files. Use the IDCAMS program to define a user VSAM data file. Figure 49 on page 294 shows example JCL that can be used to define the VSAM files.

```

//DEFVSAM JOB ...
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *

/* THE FOLLOWING SAMPLE DEFINES A */
/* VSAM INDEXED FILE */

DEFINE CLUSTER (NAME(ELA1.USER.KSDS) -
              VOL(xxxxxx) -
              CYLINDERS(pp ss) -
              KEYS(1 d) -
              RECORDSIZE(aaa mmm) -
              INDEXED)

/* THE FOLLOWING SAMPLE DEFINES A VSAM */
/* NUMBERED RELATIVE RECORD FILE */

DEFINE CLUSTER (NAME(ELA1.USER.RRDS) -
              VOL(xxxxxx) -
              CYLINDERS(pp ss) -
              RECORDSIZE(aaa mmm) -
              NUMBERED)

/* THE FOLLOWING SAMPLE DEFINES A VSAM */
/* ESDS FILE */

DEFINE CLUSTER (NAME(ELA1.USER.ESDS) -
              VOL(xxxxxx) -
              CYLINDERS(pp ss) -
              RECORDSIZE(aaa mmm) -
              NONINDEXED)

```

where:

- xxxxxx** Specifies a valid volume serial number
- pp** Specifies the primary number of cylinders to be allocated
- ss** Specifies the secondary number of cylinders to be allocated
- l** Specifies the length of the key
- d** Specifies the offset of the key
- aaa** Specifies the desired average record length
- mmm** Specifies the maximum record length

Figure 49. Defining VSAM Data Files

Defining an Alternate Index

An alternate index provides you with another way of gaining access to the records in a given KSDS file. Using a secondary key eliminates the need for you to keep several copies of the same information organized in different ways for different programs.

To gain access from an alternate index to the file through its prime index (base cluster), you must define a path to it. The path sets up an association between the alternate index and the base cluster, allowing the records in the data set to be available to you in different sequences. The alternate index is built after the base cluster is defined.

Figure 50 shows example IDCAMS definition commands for the base cluster and the alternate index cluster for an indexed file.

```
DEFINE CLUSTER (NAME(VSAM.KSDS.BASE.FILE) -  
  VOLUMES(ELAVOL) -  
  CYLINDERS(pp ss) -  
  KEYS(1 d) -  
  RECORDSIZE(aaa mmm) -  
  INDEXED)  
DEFINE ALTERNATEINDEX (NAME(VSAM.KSDS.ALT.INDEX) -  
  KEYS(1 d) -  
  CYLINDERS(pp ss) -  
  RELATE(VSAM.KSDS.BASE.FILE) -  
  VOLUMES(ELAVOL))  
DEFINE PATH(NAME(VSAM.KSDS.ALT.INDEX.PATH) -  
  PATHENTRY(VSAM.KSDS.ALT.INDEX))  
BLDINDEX INDATASET(VSAM.KSDS.BASE.FILE) -  
  OUTDATASET(VSAM.KSDS.ALT.INDEX)
```

where:

- pp** Specifies the primary number of cylinders to be allocated
- ss** Specifies the secondary number of cylinders to be allocated
- l** Specifies the key length
- d** Specifies the key displacement
- aaa** Specifies the desired average record length
- mmm** Specifies the maximum record length

Figure 50. Defining the Base Cluster and the Alternate Index Cluster

Loading Data in the Files

If you are using a VSAM indexed file (KSDS) and you want to open it for input only, initialize the file with at least one record. The file must have at least one record because a VSAM restriction prevents a file from being opened for input if the file is empty. While an empty file might be opened for output or both input and output, it must contain data to be opened for input.

There are several ways that you can put data into a file. One way is to create a VisualAge Generator program that uses an ADD process option to add records to an empty serial file. Once the program ends, you can use the IDCAMS REPRO command to copy the serial file into an indexed file.

Another way is to write a program that uses an ADD process option to add records to an empty indexed file. You must close the file in order to make the new records accessible.

Another way to initialize a VSAM KSDS file is to use a utility program shipped with the VisualAge Generator Server product. The FZEREBLK utility can be used to initialize the key of a VSAM KSDS file. Figure 51 shows how to initialize a VSAM KSDS file by setting the key to hexadecimal zeros.

```
// JOB INITIALIZE USER FILE
// DLBL KSDS,'USER.KSDS',,VSAM
// LIBDEF PHASE,SEARCH=SSSS.LLLLLL
// EXEC FZEREBLK,SIZE=AUTO
P=I,KSDS
/*
/ &
```

Figure 51. Initializing a VSAM KSDS File

where:

SSSS.LLLLLL

Specifies the name of the VSE/ESA sublibrary in which VisualAge Generator Server was installed

P=I,KSDS

Specifies that a record with a key containing hexadecimal zeros is to be written to the file with the DLBL name of KSDS

You can use the VSE DITTO utility to load data into an indexed file. Figure 52 on page 297 shows an example of loading data into a VSAM KSDS file.

```

/*
* -----
* Create temporary file TEMP.FILE and load with data to be loaded
* to the KSDS in the next step
* -----
// JOB LOAD USER FILE
// UPSI 1
// DLBL ccccccc,'vvvvvvv',,VSAM
// DLBL KSDS,'USER.KSDS',,VSAM,CAT=ccccccc
// DLBL TEMPFIO,'TEMP.FILE',0,VSAM,CAT=ccccccc,           C
//                               RECSIZE=40,DISP=(NEW,KEEP),RECORDS=(10,10)
// DLBL TEMPFII,'TEMP.FILE',0,VSAM,CAT=ccccccc,           C
//                               RECSIZE=40,DISP=(OLD,DELETE),RECORDS=(10,10)
* -----
* This first step copies the load data to the temporary sequential
* file called TEMPFIO.
* -----
// EXEC DITTO
$$DITTO CSQ FILEOUT=TEMPFIO,CISIZE=2048
0001Employee1 Address1
/*
* -----
* This second step copies the data in the SAM file to the KSDS
* -----
$$DITTO SVS FILEOUT=KSDS,FILEIN=TEMPFII,RECSIZE=40,NLRECS=1
$$DITTO EOJ
/*
/ &
* $$ EOJ

```

Figure 52. Loading a VSAM KSDS File

where:

ccccccc

Specifies the file name of the catalog

vvvvvvv

Specifies the file identifier of the catalog

Support for DBCS Terminals

VisualAge Generator Server provides support for the IBM Personal System/55 and the IBM 5550 family of terminals (emulating an IBM 3270 device). In addition to the basic hardware, this support uses character set F8 and four hardware attributes for double-byte character set (DBCS). The extended attributes are shift-out (SO) and shift-in (SI) enable, field outlining, color, and extended highlighting.

For the MVS CICS, MVS/TSO, VSE, and VM CMS environments, VisualAge Generator Server sends hardware attributes to the terminal only if the terminal supports them. The attributes are ignored if the terminal does not support them.

Library Space Considerations

When VisualAge Generator programs are generated on the workstation, several files are uploaded and cataloged on VSE/ESA. These files include a parts control file (part type PCF), COBOL source (part type C), an object module (part type OBJ), as well as the phase (part type PHASE). In some cases, users might choose to delete some of these files from the library. When parts are deleted, the space occupied is normally “freed-up”. However, this is not true for libraries and sublibraries that are concurrently shared between partitions on a single VSE/ESA system, or that are shared between different VSE/ESA systems.

For these libraries and sublibraries, the space is released only when they are in a non-shared status. It is important to periodically release space in these libraries to avoid receiving a “Library Full” condition. By running the sample JCL shown in Figure 53, you can force the release of space for these shared libraries and sublibraries. All space previously occupied by deleted parts will be freed up.

Note: For additional information on the LIBR RELEASE command, see the *VSE/ESA System Control Statements Manual (SC33-6513)*.

```
* $$ JOB JNM=RELS,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB RELS  RELEASE LIBRARY SPACE
// EXEC LIBR
RELEASE SPACE SUBLIB=PRD2.EZELIB
/*
/&
* $$ EOJ
```

Figure 53. Sample VSE/ESA JCL to Release VSE/ESA Library Space

Extended Addressing Considerations for VisualAge Generator Server

Most of the code shipped with VisualAge Generator Server runs in 31-bit addressing mode and resides above the 16MB line.

Most of the storage acquired by VisualAge Generator Server is above the 16MB line unless the first VisualAge Generator-generated program in the run

unit is link-edited with AMODE(24) or is compiled with the /DATA=24 generation option. The AMODE(24) program attribute specifies that the program runs in 24-bit addressing mode. Refer to the *VisualAge Generator Generation Guide* for more information on AMODE(24) or the /DATA generation option.

VisualAge Generator programs are capable of running in extended addressing mode. The complete VisualAge Generator-generated COBOL program under VSE CICS and VisualAge Generator Server can run above the 16MB line on systems supporting 31-bit operation. VisualAge Generator programs reduce the contention for virtual storage below the 16MB line, providing virtual storage constraint relief (VSCR). This VSCR is especially significant under VSE CICS.

On systems where 31-bit addressing mode is not supported, both the generated COBOL program and the VisualAge Generator Server program run in 24-bit addressing mode.

If your system runs in ESA mode, VisualAge Generator Server generally runs in 31-bit addressing mode even if the generated program is link-edited as a program with a 24-bit addressing mode. Under both VSE CICS and VSE batch, most of the VisualAge Generator Server modules are link-edited with the AMODE(31) and the RMODE(ANY) program attributes. The AMODE(31) program attribute specifies that the program is prepared to handle a 31-bit address length on entry. The RMODE(ANY) program attribute specifies that the program can reside at any virtual storage location, either above or below the 16MB line.

Most storage requests are made from "above the line" if the system runs in 31-bit addressing mode. Some exceptions to the code running in 31-bit addressing mode are detailed below:

- In VSE CICS, DL/I requests are performed in 24-bit addressing mode. DL/I call parameters and I/O buffers are located below the 16MB line.
- In VSE batch, VSE/POWER requests are performed in 24-bit addressing mode. The storage needed for those requests is obtained from storage areas below the 16MB line.
- In VSE batch, DTF I/O requests are performed in 24-bit addressing mode. The storage needed for those requests is obtained from storage areas below the 16MB line. These service requests include writing ELA messages to the system logical unit SYSLST, routing dumps to the system logical unit SYSLST, reading trace options from the system logical unit SYSIPT, and reading SID parameters from the system logical unit SYSIPT for SQL programs.

If your program calls programs that need to run in 24-bit addressing mode then the program must be generated with the /DATA=24 generation option and linked with the AMODE(24) program attribute. Any table program or mapping services program must be linked with AMODE(31) or RMODE(ANY), and generated with the /DATA=24 generation option.

Note: On CICS, the /DATA=24 generation option can be used only if the dynamic storage required by the generated COBOL program, print services program, or table is less than 64KB (KB equals 1024 bytes) in length. This refers to dynamic storage as defined in the COBOL working storage section.

Database Considerations

This section discusses preparing programs and checking access authorization to database resources when doing the following:

- Using DB2 on MVS systems
- Using DB2/VSE on VSE systems and moving program packages among VSE systems
- Using SQL on VM systems

Preparing Programs

Before running a program, the SQL* statements need to be analyzed and prepared. You can choose to have your programs prepared by DB2 (the default on MVS systems), DB2/VSE (the default on VSE and VM systems), or an ANSI SQL database manager.

If you specify the /ANSISQL generation option, any SQL statements in the generated program are generated using the ANSI SQL format. If you use ANSI SQL, you need to tailor the templates, procedures, and execs for your database manager.

If you use DB2, you also need to bind the DB2 program plan. Refer to the [document](#) for information on binding DB2 program plans.

If your programs run in the MVS/TSO, MVS batch, or IMS BMP environments, you might also need to tailor the runtime JCL and CLIST templates. If your programs run in the VM CMS or VM batch environment, you might also need to tailor the runtime exec templates. Refer to the *VisualAge Generator Generation Guide* for additional information on tailoring runtime JCL, CLIST, and exec templates.

Checking Access Authorization

The database manager checks whether program users have the authority to access tables or run programs. The type of checking done varies depending on your system and the processing mode.

To run a DB2/VSE program, program users must be granted authority to run the program and to process any tables accessed using dynamic SQL statements in the program.

Moving Program Packages among VSE Systems

If you have generated an SQL program and run the preparation job or REXX exec created by VisualAge Generator Developer, your program and the SQL package that is created during the DB2/VSE preprocessing step contain consistency tokens based on a time stamp. The program and its associated SQL package must have consistency tokens that match.

On VSE systems, you must ensure that your program and its SQL package are created in the same preparation job stream.

On VM systems, you must ensure that the program module and the SQL package you are using were created by the same invocation of the preparation REXX exec. Refer to the CTOKEN parameter description of the SQLPREP command in the *DB2/VSE Application Programming for IBM VM Systems* manual for more information.

If you need to run your program on a different VSE system than the one for which it was generated, do the following:

1. Use the VSE Librarian BACKUP/RESTORE function to move the program from the VSE sublibrary where it resides to the VSE sublibrary on the VSE system where you plan to run your program.

Note: This step is optional because it is possible to use the same load library for different VSE systems.

2. Use the DB2/VSE DBS utility, UNLOAD PACKAGE, to copy the SQL package to a sequential file. Then use the RELOAD PACKAGE utility to read the package back into the database server on the VSE system where you intend to run your program. Refer to the DB2/VSE database services utility for VSE for more information on unloading and reloading packages.

Fold Tables

VisualAge Generator Server does folding (uppercase translation) using a 256-byte translate table. VisualAge Generator Server uses the translate table at offset X'0C' in CSECT ELANCccc where ccc is the language code.

Backing Up Data

You should regularly back up your data. This includes all files related to VisualAge Generator Server, private libraries, user-created data files, and user load libraries. System services are provided to back up and restore user libraries.

Customizing VisualAge Generator Server

Customizing VisualAge Generator Server consists of performing some of the same procedures used to install the product on the system. These procedures are described in the *Program Directory for VisualAge Generator Server* for your MVS, VSE or VM system. The program directory contains information on changing system options.

Chapter 31. System Considerations for VSE CICS

This chapter provides additional system requirements and considerations for administering VisualAge Generator Server in the VSE CICS environment.

The following information is discussed:

- Required file descriptions
- Segmented and nonsegmented processing
- Printing considerations
- Using the New Copy function
- Recovery and database integrity considerations
- Considerations that affect performance
- Using and allocating data files
- Considerations for using DB2/VSE
- Considerations for using DL/I

Required File Descriptions

VisualAge Generator Server requires the following files:

File Name

Description

ELAD This transient data queue is the default destination for VisualAge Generator Server error messages. VisualAge Generator Server produces error messages when it detects an error that prevents a program from continuing.

The ELAD transient data queue is defined when VisualAge Generator Server is installed. If you want to direct error messages for different transactions to different queues, define the other queues with the same characteristics as ELAD. Use the error diagnostic utility ELAC to direct error messages to the required queue. See the description of the utility in “Chapter 39. Diagnostic Control Options” on page 383 for more information.

EZEPRINT

The file that you associate to the VisualAge Generator file name EZEPRINT at resource association will be used when printing from a program that displays printer maps. This file can be defined with a file type of SPOOL or TRANSIENT. This file is normally associated with the transient data queue EZEP.

If you installed VisualAge Generator Server as described in the VisualAge Generator Server program directory, EZEP is defined as an indirect destination associated with the system printer. The maximum record length that a generated program writes to the system printer is 650 bytes for double-byte character set (DBCS) print maps and 133 bytes for single-byte character set (SBCS) print maps. The first byte is an American National Standards printer control character. The DBCS record length is longer than the physical printer line length because the print record can contain outlining and shift-out/shift-in (SO/SI) control characters that do not appear on the device.

If you are using VisualAge Generator Server to print to a file destination other than EZEP, the characteristics of that file should be the same as EZEPRINT. The file destination can be specified either during resource association at generation time or with the EZEDESTP special function word at run time.

EZEPRMG

This VSAM indexed file (KSDS) contains the parameter group records used for print control options for the VisualAge Generator Server terminal printer utility, FZETPRT. The FZETPRT program reads this file searching for the parameter group matching the transaction name that started FZETPRT. See “Special Parameter Group for the FZETPRT Program” on page 307 for a description of the print parameters. See “Chapter 41. Using the Parameter Group Utility” on page 393 for more information about maintaining this special parameter group.

Segmented and Nonsegmented Processing

Generated programs can run in nonsegmented (CICS conversational) mode or segmented (CICS pseudoconversational) mode, and can switch between the two modes. When a program converses a map, a pseudoconversational terminal I/O operation is used if any of the following is true:

- The EZESEGM special function word is set to 1.
- The program issues an XFER statement with a map.
- The program was specified to run in segmented mode and the value was not overridden with the EZESEGM special function word.

The current transaction ends and the program status is saved in a temporary storage queue until the terminal input is received.

Two different storage queues are used. The storage queue names have the following format:

xyyytttt

where:

- x** Specifies a byte with the hex value X'EE'
- yyy** Specifies WRK (program working storage) or MSG (current map saved across help or error display)
- tttt** Specifies the terminal ID associated with the transaction

The /WORKDB generation option specifies whether a main or auxiliary storage queue is used. The storage queues are deleted at the end of a run unit.

Refer to the *VisualAge Generator Design Guide* for more information on developing segmented programs.

Using Transient Data Queues for Printing in VSE CICS

When a program displays a printer map, the printed output is sent to the logical file named EZEPRINT. The resource association file used by generation specifies whether the EZEPRINT file is a spool file or a transient data queue. Refer to the *VisualAge Generator Design Guide* for more information on using spool files for printing.

Processing a DISPLAY Process Option for a Printer Map

When printing is initiated through the DISPLAY process option for a printer map, the printer output is routed to the file that is specified as the resource associated to the EZEPRINT data set. The resource associated to EZEPRINT can be specified during resource association at generation time, or can be specified at run time using the EZEDESTP special function word. The remainder of this section describes how to set up transient data queues for printed output.

Printed output destined for a transient data queue is accumulated in temporary storage. The temporary storage queue name has the following format:

ttttnnn

where:

- tttt** Is the transient data queue name
- nnnn** Is the EXEC Interface Block (EIB) task number

When a program ends, or a CLOSE process option for a print map is issued, or a segmentation break occurs, VisualAge Generator Server enqueues on a transient data queue to prevent interspersed printing from other transactions. VisualAge Generator Server copies the printed output onto the transient data

queue. The printed output is in line character format with an American National Standards printer-control character.

Specifying Print Destination

The default print destination in MVS CICS, VSE CICS, and VSE batch is a transient data queue named EZEP. If you installed VisualAge Generator Server as described in the VisualAge Generator Server program directory, EZEP is an indirect destination associated with the system printer. During program generation, this destination can be changed to any 4-character transient data queue name. The destination control table (DCT) entry for the queue determines the actual destination. The destination can be the system printer, a data set, or a terminal printer.

You can override the default destination at generation time by specifying the alternate destination as the system resource name for the EZEPRINT file. You can change the print destination at run time by using the EZEDESTP special function word. Refer to the VisualAge Generator online help system within VisualAge Generator Developer for additional information on the EZEDESTP special function word.

VisualAge Generator Server also provides a way of starting an asynchronous print task from a program and controlling the print destination from the program starting the asynchronous task. To do this, define the print task as a batch program and generate it with the option /PRINTDEST=TERMID. Use a CALL CREATX statement to start the batch program, specifying the print destination in the CREATX service routine parameters. The program ignores the generated print destination and uses the destination specified in the CREATX service routine. Refer to the VisualAge Generator online help system within VisualAge Generator Developer for more information on the CREATX service routine.

CICS Terminal Printing

The program called FZETPRT supports terminal printing. This program runs as a CICS transaction that starts automatically when records are written to the transient data queue. If VisualAge Generator Server was installed as described in the VisualAge Generator Server program directory, the transaction name is EZEZ for IBM 5550-type printers and EZEP for all other printers. To send printed output to the terminal, you must include a TYPE=INTRA for the transient data queue in the CICS destination control table (DCT). Specify EZEP or EZEZ for the transaction ID in the DCT entry. The trigger level in that entry must be set to 1 to ensure proper output. See “Printing Transient Data at a Terminal Device” on page 72 for a sample DCT entry.

When the FZETPRT program is initiated, it reads a line from the transient data queue, converts the American National Standards printer-control

character to NL EOM format, and writes to the terminal printer specified in the DCT entry. If you do not specify a terminal name in the DCT entry, the queue identifier must be the same as the terminal printer identifier. The FZETPRT program buffers multiple print lines into a single CICS SEND command to improve performance.

When using terminal printing with VisualAge Generator Server, you should be aware of potential problems regarding form-feed orders and page alignment. When the FZETPRT program is triggered, a form-feed order is issued to the printer to ensure that it begins printing at the top of a page. If a second map is sent to the queue before it is emptied by the FZETPRT program, a form-feed order is not issued before the second map is printed. Page alignment can vary depending on the timing with which successive maps are sent to the queue.

Another potential problem can occur when printing successive maps. If one of the maps in the series is defined with lines equal to, or one line fewer than, the lines-per-page setting on the printer, a blank page occurs between the printed maps. To avoid this, define the map size as 2 lines fewer than the lines-per-page setting on the printer. Because the FZETPRT program inserts a new-line order to ensure that printing begins in column 1, the first line of the map to be printed is actually printed on the second line of the page. The second line must be allowed because a new-line order is added after the last line of the map, which advances the print head to the beginning of the next line. If this happens to be the first line of the following page, the next form-feed order causes the page to be skipped before printing resumes.

Another thing to consider is that although VisualAge Generator Server sometimes causes successive, stand-alone form-feed orders (“1”), the FZETPRT program suppresses all but one of these in converting them to NL EOM format.

If these form-feed considerations are too restrictive for your needs, consider using the FORMFD=NO parameter described in “FORMFD Parameter” on page 62 .

Special Parameter Group for the FZETPRT Program

You can provide terminal printing parameters to the FZETPRT program to vary the printed output by using a special parameter group file.

The FZETPRT program attempts to read a file named EZEPRMG for a parameter group that has the same name as the transaction used to start the FZETPRT program. For example, if the print transaction that starts the FZETPRT program is named EZEP, then FZETPRT tries to find the parameter group named EZEP. If the parameter group is not located in a file named

EZEPRMG, or if EZEPRMG does not exist, then the FZETPRT program reads the DCAPRMG file to find the parameter group associated with this transaction.

When the transaction starts, the FZETPRT program reads the parameter group and varies the printer output according to the contents. If you need to use the terminal printing parameters, create a parameter group using the VisualAge Generator Server utility provided for this purpose. See “Chapter 41. Using the Parameter Group Utility” on page 393 for more information.

For this parameter group, you can specify the following four parameters:

- PRTBUF=xxx
- PRTMPP=nnn
- PRTTYP=D
- FORMFD=NO

Note: Do not include blanks between keywords and their associated values.

PRTBUF Parameter

Use the PRTBUF parameter to set the size of the printer buffer. The number of SEND commands sent to the terminal printer depends on the size of the printer buffer. The following example shows how to specify the buffer size using the PRTBUF parameter:

```
PRTBUF=xxx
```

where:

xxx Is the size in bytes of the printer buffer

The FZETPRT program uses a default buffer size if any of the following conditions occur:

- The parameter is not specified in the parameter group.
- There is no parameter group associated with the transaction.
- The parameter keyword is misspelled.
- The value specified is not valid (values greater than 8K bytes, smaller than 480 bytes, or not numeric).
- The EZEPRMG or DCAPRMG file does not exist or is not available.

The default buffer size is 2KB (where KB equals 1024 bytes) for the standard character set printers and 480 bytes for LU type 3 printers.

For double-byte character set (DBCS) users the default buffer size and the maximum buffer size allowed is 1918 bytes. The default value is used if your specified value exceeds the maximum number of bytes.

When the buffer size is larger than the default, usage of the PRTBUF parameter is optional. However, using the PRTBUF parameter is recommended to reduce the number of SEND commands sent to the terminal. If the printer buffer size is smaller than the default, specify the real buffer size using this parameter. Not specifying the real buffer size can cause unpredictable results.

PRTMPP Parameter

Use the PRTMPP parameter to set the maximum number of print positions. The following example shows how to specify the number of print positions using the PRTMPP parameter:

```
PRTMPP=nnn
```

where:

nnn Is the physical length (maximum print position) of the printer line

The FZETPRT program assumes a default maximum print positions of 132 if any of the following occurs:

- The parameter is not specified in the parameter group.
- There is no parameter group associated with the transaction.
- The parameter keyword is misspelled.
- The value specified is not valid (not numeric).
- The EZEPRMG or DCAPRMG file does not exist or is not available.

Use caution when coding the value of this parameter. If the value entered is a valid numeric, the FZETPRT program uses the value without validating it. If the value is greater than the number of print positions available on the actual printer, possible malfunctioning can take place causing more line skips than necessary.

Note: For DBCS users, this parameter must be specified unless the printer is configured with MPP=132.

PRTTYP Parameter

Use the PRTTYP parameter if you use a DBCS printer. The following example shows how to specify the use of a DBCS printer using the PRTTYP parameter:

```
PRTTYP=D
```

Note: This parameter must be used to specify that you are a DBCS user and your output is being directed to an IBM 5550-family printer.

If you use multiple printers with different characteristics (namely different MPP, different buffer size, or DBCS versus non-DBCS printers), you need as many transaction IDs as there are printers, each one associated with the FZETPRT program. For examples of table entries for two printers, see the CICS transaction definitions provided with VisualAge Generator Server for the EZEP (non-DBCS printers) and EZEZ (DBCS printers) transactions.

FORMFD Parameter

Use the FORMFD parameter to control the form-feed orders that the FZETPRT program issues. The following example shows the format of the FORMFD parameter:

```
FORMFD=NO
```

The FZETPRT program defaults to inserting form-feed orders into the printer data stream if any of the following occurs:

- The parameter is not specified in the parameter group.
- There is no parameter group associated with the transaction.
- The parameter does not appear as FORMFD=NO.
- The EZEPRMG or DCAPRMG file does not exist or is not available.

If the parameter is specified correctly, the FZETPRT program does not insert form-feed orders for any reason. This includes issuing a SET map PAGE statement, closing the printer, or the initial form feed that is normally done. All forms control depends on the map size specified during map definition.

CICS Entries for FZETPRT (DBCS only)

If you are using an SCS-type printer and you use DBCS, ensure that your system programmer has coded the destination control table (DCT) and the program control table (PCT) entries for a transaction that runs FZETPRT with the following option:

```
MSGPOPT=CCONTRL
```

The MSGPOPT option defines the optional facilities that a task can use. The CCONTRL parameter indicates that the program can control the outbound chaining of request units. Refer to the CICS manuals for more information.

Using the New Copy Function

The new copy function (either the VisualAge Generator Server new copy utility or the CICS NEWCOPY command) causes a transaction to use a new copy of a program, map group, or table referenced in the transaction. The VisualAge Generator Server new copy utility is implemented as a VisualAge Generator program in the CICS environment. Active transactions continue to use the current version of a program, map group, or table until the transaction either completes or reaches the end of a segment. A new copy of the program, map group, or table is then made available to the transaction by VisualAge Generator Server. Use the new copy function when programs, map groups, and tables are modified and generated again. This enables you to install new versions of programs, map groups, and tables onto your system without disrupting operation.

For programs and map groups, you can use the CICS NEWCOPY command or the VisualAge Generator Server new copy utility to cause the new copy of the program to be used the next time a load request is issued for the program. The VisualAge Generator Server new copy utility does a new copy for both the online print services program and the map group format module when you specify a part type of Map Group. If you use the CICS NEWCOPY command for a map group, you must issue the NEWCOPY for both the online print services program and the map group format module.

For tables, you must use the VisualAge Generator Server new copy utility to cause a fresh copy of the table to be used the next time a load request is issued for the table. Do not use the CICS NEWCOPY command for tables. The VisualAge Generator Server new copy utility sets a flag indicating that the new copy of the table is to be used the next time a program loads the table contents.

For more information on the VisualAge Generator Server new copy utility, see “New Copy” on page 377.

Recovery and Database Integrity Considerations

This section presents suggestions for specifying recovery options in CICS tables, and presents a discussion of the concept of a logical unit of work (LUW) that can help you determine your database recovery and integrity options.

VisualAge Generator programs can make use of all the MVS CICS recovery and data integrity features. Refer to the recovery and restart information for your release of CICS for a description of these features.

Specifying Recovery Options in the CICS Tables

The system initialization table (SIT) for CICS should specify DBP=XX, where XX is not equal to NO. If the DBP value is not equal to NO it prevents ASPE abends when generated programs issue CICS SYNCPOINT and CICS SYNCPOINT ROLLBACK commands.

If DTB=YES is specified on the program control table (PCT) entries for the transactions, the value specified for DBP is significant. CICS provides two dynamic backout programs, one for systems that require DL/I support and the other for systems that do not require DL/I support. These programs are provided by CICS if an entry is included in the processing program table (PPT) that specifies TYPE=GROUP and FN=BACKOUT.

Determining When Database Changes Are Committed

By understanding the concept of a logical unit of work (LUW) in VisualAge Generator programs, you are able to determine when changes are committed to your database. This section presents a definition of an LUW and some considerations for where to place commit points in your programs.

Using CICS recovery support, you can define your files and databases to be recoverable resources. If your program makes changes to a recoverable resource, the changes are not committed to the file or database until the end of an LUW. If your program ends abnormally before reaching the end of an LUW, all changes that were made since the beginning of the LUW are backed out.

An LUW for a transaction ends whenever a commit point or a rollback occurs. A commit point occurs when one of the following occurs:

- The run unit ends successfully. In MVS CICS, VSE CICS, and VSE batch, a run unit is equivalent to a single transaction and consists of all VisualAge Generator programs and non-VisualAge Generator programs that transfer control among themselves using a DXFR or CALL statement. For non-VisualAge Generator programs, this also includes any transfer that uses a CALL statement, CICS LINK command, or CICS XCTL command.
- A program uses a CONVERSE process option, and any of the following is set to 1:
 - EZESEGM special function word (segmented mode)
The EZESEGM special function word defaults to 1 if the program is defined as segmented.
 - EZECNVCM special function word (commit at every CONVERSE process option)
 - EZEDLTRM special function word (end the PSB at CONVERSE process option) if the program uses DL/I

These options cause the commit to be issued after terminal output and before the next terminal input. A commit point at terminal I/O synchronizes the updates to the database and confirmation messages to the program user.

- A transfer using an XFER statement occurs.
- A program calls either the EZECOMIT or COMMIT service routine.
- A transfer using a DXFR statement occurs and a PSB is scheduled, and one of the following occurs:
 - A transfer to a non-VisualAge Generator program and a PSB is scheduled
 - The /SYNCDXFR generation option was specified for the transferred-from program
 - The /NOSYNCDXFR generation option was specified for the transferred-from program and different PSB names were identified in the program specifications for the two programs
- A called DL/I program returns to the calling program, the PSB was not passed using the EZEDLPSB special function word, and PCBs were not passed using the EZEDLPCB special function word.

A rollback occurs when one of the following occurs:

- A program calls the EZEROLLB or RESET service routine.
- A program ends because of an error condition.

When a rollback occurs, all changes that were made to a database and recoverable files since the start of the LUW are backed out.

Considerations that Affect Performance

This section describes factors that affect system performance and suggestions on how to improve performance.

Residency (Phases in Memory) Considerations

The performance of a program is affected by the number of times that a running program requires access to a disk. Programs require access to disks for the following reasons:

- Locating and loading VisualAge Generator Server phases
- Retrieving and storing user data
- Locating and loading program, map group format modules, online print services programs, and table phases

VisualAge Generator Server loads objects as they are needed. For example, VisualAge Generator Server loads a program, online print services program,

map group format module, or table when another program calls or references it. If you make an object resident, then the object remains in storage after it is loaded by VisualAge Generator Server. You can use the resident parameter on the PPT entry to make a program, map, or online print services program resident.

For tables, the Shared, Resident, and Keep after use options are used to control residency. The Shared, Resident, and Keep after use options are definition attributes in VisualAge Generator Developer. The Shared and Resident options are specified within table definition and the Keep after use option is specified within program definition when specifying tables and additional records. These attributes cannot be overridden at generation time. If these attributes need to be changed, the definition must change. After generation, the change takes effect for all programs that contain the respective tables.

Virtual Storage Considerations and Residency

It is true that if a program, online print services program, map group format module, or table is resident, less I/O is required for multiple loads. However, making these objects resident requires more virtual storage because the modules accumulate in storage as they are loaded and are not deleted after they are used.

When deciding what to make resident, consider the following:

- Storage constraints
- Frequency of program use
- Long running programs versus programs that are started more frequently

Because most systems have virtual storage constraints, it is not possible to make everything resident. You should establish priorities for deciding which objects you want to make resident. These residency priorities reflect a trade-off between program usage and storage constraints. Your priorities can dictate that some components of a program (such as the online print services program and the map group format module) must be made resident, while other components (such as tables) must not be made resident.

Generally, objects that are loaded more than once are prime candidates for residency. Examples of this are a table that is used by more than one program or a program that is called more than once.

The following are some general rules for residency:

- When deciding what to make resident, consider the following objects:
 - Called programs
 - Online print services module

- Map group format modules
- Tables
- Main programs
- Programs that are started or referenced frequently should be made resident. This includes programs that are started by VisualAge Generator CALL or DXFR statements.
- To prevent VisualAge Generator Server from deleting a table after it is used, you should specify the KEEP AFTER USE option in program definition when specifying tables and additional records. In addition, if you want to make a table resident, use Resident and Shared within table definition.
- Programs that are not frequently initiated or have long running time should not be made resident.

In VSE CICS, when a program component is made resident, it remains in storage from the time it is loaded into storage until either CICS is shut down or the new copy function is used. To aid in deciding which programs should be made resident, you can use CICS shutdown statistics to determine how often a generated program or other component is loaded into the partition.

Pseudoconversational Programs and Residency

In VSE CICS, if you plan to run a program in pseudoconversational mode, you should consider making all components of the program resident. In pseudoconversational mode, the program and its components are deleted and loaded again at each segment break (for example, at each segmented CONVERSE process option) if they are not made resident. This degrades the performance of pseudoconversational mode.

Work Database Temporary Storage Queue Considerations

When running in pseudoconversational mode (using a segmented CONVERSE process option), the data and the status associated with the program must be saved during user think time. You can control whether this information is saved into the CICS main temporary storage or auxiliary storage. Using main temporary storage can result in better performance because the data is written to memory within the CICS address space instead of writing the data to disk space.

The /WORKDB generation option can be used to choose whether main or auxiliary temporary storage is used for the work database. See the *VisualAge Generator Generation Guide* for more information about specifying this option.

Note: Use of main temporary storage can degrade system performance because the increased address space that is referenced can increase the paging activity. Also, CICS can experience a short-on-storage condition

if the program data to be saved exceeds the available CICS storage. Therefore, if you take advantage of main temporary storage for programs requiring better performance, you should monitor your system to ensure that virtual storage problems do not occur.

The amount of data written or read on each request to CICS when saving program data and status, can also affect performance. The installation options module, ELARPIOP, specifies the largest size record VisualAge Generator Server writes to main or auxiliary temporary storage. The default size is 32KB (where KB equals 1024 bytes), which is the largest value allowed by CICS. Use a large value to ensure that the least number of write requests are required, and, if using auxiliary storage, to ensure that the least number of I/O operations are required. See the *VisualAge Generator Server Program Directory* for information on how to change the value in the installation options module.

Note: If you are using auxiliary storage queues, you should ensure the control interval size (CISIZE) of the VSAM data set used for auxiliary temporary storage matches the size specified in the installation options file. If the CISIZE for the data set is smaller, CICS splits the data written or read into smaller pieces and does multiple I/O operations for each VisualAge Generator Server request. Also ensure that you have an adequate number of buffers for the auxiliary temporary storage data set in order to reduce the number of physical I/O operations.

Conversational and Pseudoconversational Processing

Conversational processing can be more efficient than pseudoconversational. If storage and other CICS resources are not a constraint on an XA or ESA system, define programs as nonsegmented, and set the EZECNVCM special function word in the program to 1. Setting the EZECNVCM special function word to 1 causes file and database resources to be released for each CONVERSE process option. This combines the better performance of nonsegmented processing with the resource-sharing characteristics of segmented processing.

Terminal Printing

The performance of terminal printing can be enhanced by specifying the PRTBUF parameter for the FZETPRT program. See “MVS CICS Terminal Printing” on page 58 for more information on terminal printing and the PRTBUF parameter.

Monitoring and Tuning the System

For VSE CICS, you can use the CICS/VSE Performance Analysis Reporting System (CICSPARS/VSE) to monitor performance. This tool provides information about current tasks, snapshots of the current system status, and information about the VSE CICS dynamic storage area, as well as the ability to display VSE CICS storage.

Refer to the performance guide for your release of VSE CICS for more information.

Using and Allocating Data Files in MVS CICS and VSE CICS

This section describes how to define data files for use in generated VisualAge Generator programs in the MVS CICS or VSE CICS environment.

Defining and Loading VSAM Data Files

Before CICS programs can use VSAM data files, you must define and load them. See “Defining and Loading VSAM Program Data Files” on page 292 for information on defining VSAM data sets, defining an alternate index, and loading a VSAM data set.

Adding the Job Control Statements

After the data set has been defined and loaded, add the data set name to the CICS startup JCL or the VSE/ESA standard label area to allocate user files. You can also let MVS CICS dynamically allocate the data set to the file using the information specified in the file control table (FCT). Figure 54 shows example allocation statements for an indexed, relative, and serial file, and an alternate index.

```
// DLBL USERCAT, 'USERCAT', ,VSAM
// DLBL KSDS, 'ELA1.USER.KSDS', ,VSAM, CAT=USERCAT
// DLBL KSDSAIX, 'ELA1.USER.KSDS', ,VSAM, CAT=USERCAT
// DLBL RRDS, 'ELA1.USER.RRDS', ,VSAM, CAT=USERCAT
// DLBL ESDS, 'ELA1.USER.ESDS', ,VSAM, CAT=USERCAT
```

Figure 54. Allocating User Files

Adding the File Name to the CICS File Control Table

After the data set has been defined, loaded, and added to the CICS startup JCL, the FCT entry must be created for the file name for a CICS program to gain access to the data set. Creating an FCT entry can be accomplished using online (RDO) or macro definitions.

Figure 55 on page 319 shows sample macro definition entries that can be used to add a file name to the FCT. VisualAge Generator Server uses the name on the FILE operand. The FILE operand name must be the same as the ddname (MVS) or the DLBL name (VSE) in the CICS startup JCL. All other operands must be the same as when you add an indexed, relative, or serial file to the FCT.

With MVS CICS or VSE CICS, make an entry to the FCT for every file used by a program. The CICS files can be defined as remote FCT entries.

For further information, refer to the appropriate CICS resource definition guide for your environment.


```

*** THIS IS A SAMPLE OF AN FCT ENTRY FOR A KEY-SEQUENCED ****
*** DATA SET(KSDS). ****
    DFHFCT TYPE=FILE, C
        FILE=KSDSFILE, C
        ACCMETH=VSAM, C
        SERVREQ=(READ,UPDATE,ADD,DELETE,BROWSE), C
        FILSTAT=(ENABLED,CLOSED), C
        RECFORM=FIXED
*** END OF KSDS *****
*** THIS IS A SAMPLE OF AN FCT ENTRY FOR AN ALTERNATE INDEX **
*** DATA SET(KSDS). ****
    DFHFCT TYPE=FILE, C
        FILE=KSDSAIX, C
        ACCMETH=VSAM, C
        BASE=KSDS, C
        SERVREQ=(READ,UPDATE,ADD,DELETE,BROWSE), C
        FILSTAT=(ENABLED,CLOSED), C
        RECFORM=FIXED
*** END OF KSDS *****
*** THIS IS A SAMPLE OF AN FCT ENTRY FOR A RELATIVE RECORD ***
*** DATA SET(RRDS). ****
    DFHFCT TYPE=FILE, C
        FILE=RRDSFILE, C
        ACCMETH=VSAM, C
        SERVREQ=(READ,UPDATE,ADD,DELETE,BROWSE), C
        FILSTAT=(ENABLED,CLOSED), C
        RECFORM=FIXED
*** END OF RRDS *****
*** THIS IS A SAMPLE OF AN FCT ENTRY FOR AN ENTRY-SEQUENCED ***
*** DATA SET(ESDS). ****
    DFHFCT TYPE=FILE, C
        FILE=ESDSFILE, C
        ACCMETH=VSAM, C
        SERVREQ=(READ,UPDATE,ADD,DELETE,BROWSE), C
        FILSTAT=(ENABLED,CLOSED), C
        RECFORM=FIXED
*** END OF ESDS *****
***** END OF USER DATA FILES *****

```

Figure 55. Adding the File Name to the FCT

Using Remote Files

VisualAge Generator programs can gain access to files that do not reside on your MVS CICS or VSE CICS system. Refer to the *VisualAge Generator Client/Server Communications Guide* for more information. Refer to the section on developing client/server programs in the *VisualAge Generator Generation Guide* for information on gaining access to remote files from generated programs on either a workstation or a CICS host session. Also refer to the *VisualAge Generator Generation Guide* for information on specifying linkage tables. Refer to the appropriate CICS manuals for information about defining remote programs, transactions, or files.

Defining Transient Data Queues

Transient data queues are used in MVS CICS and VSE CICS for reading or writing data from tapes, disks, or other sequential (MVS) or serial (VSE) files. If you associated a serial or print file with a transient data queue at generation, you must define a MVS CICS or VSE CICS destination control table (DCT) entry for the queue.

You can define the following types of transient data queues:

- Intrapartition (temporary data)
- Extrapartition (data that other non-CICS regions can use)

Intrapartition transient data files contain data that is not usable after it is read.

Defining Intrapartition Transient Data

The following two examples show how to define intrapartition transient data files.

Passing Transient Data between CICS Transactions: This is an example of a DCT entry that can be used to pass data from one CICS transaction to another. The file destination specified at generation with the SYSNAME option should be xxxx.

```
DFHDCT TYPE=INTRA,           C
      DESTID=xxxx,           C
      DESTFAC=FILE
```

Printing Transient Data at a Terminal Device:

This is an example of a DCT entry that can be used for terminal printing in VisualAge Generator Server. At generation, the print destination for EZEPRINT can be specified. The default is EZEP. (A DCT entry is supplied for EZEP that sends the printed output to the system printer.) The program supplied for printing, FZETPRT, reads records from the transient data queue and issues SEND commands to the terminal in order to print the records.

In this sample DCT, the PR01 terminal is to receive the printed output. PR01 is a MVS CICS or VSE CICS printer terminal name. You specify the EZEPRINT destination at generation as PR01. VisualAge Generator Server writes the printed output to the transient data queue, PR01. The transaction EZEP starts and causes the program FZETPRT to run. The data is read from the transient data queue and sent to the terminal, PR01. The program control table (PCT) entries for EZEP and the processing program table (PPT) entries for FZETPRT are supplied. You must supply the destination control table and the terminal control table entries for the transient data and terminal.

```

DFHDCT TYPE=INTRA,           C
      DESTID=PR01,           C
      DESTFAC=TERMINAL,      C
      TRANSID=EZEP,          C
      TRIGLEV=1

```

If the terminal printer is a DBCS printer, specify EZEZ as the TRANSID.

Defining Extrapartition Transient Data

Data to be read from tape or sent to a printer is contained in extrapartition transient data queues.

The following two examples show how to use extrapartition transient data queues. These files can be used by non-CICS devices and by CICS.

Printing Transient Data: This is an example of a DCT entry specification that can be used to print output on a high-speed system printer. The file destination specified at generation with the SYSNAME option should be *zzzz*.

You need to add the appropriate VSE JCL and VSE/POWER JECL to the CICS processing JCL to assign a printer to the system number, SYS005. The following sample entry for the DCT is for printed output with an American National Standards printer- control character:

```

DFHDCT TYPE=EXTRA,           C
      DESTID=ZZZZ,           C
      DSCNAME=PRINTER
DFHDCT TYPE=SDSCI,           C
      DSCNAME=PRINTER,       C
      RECFORM=FIXUNB,        C
      RECSIZE=121,           C
      TYPEFLE=OUTPUT,        C
      DEVICE=1403,           C
      CLTCHR=ASA,            C
      DEVADDR=SYS005

```

Reading Transient Data from a Tape: This is an example of a DCT entry specification that can be used to read data from a tape. The VisualAge Generator Developer definition for this record specifies the file as a serial file with a file ID of *yyyy*.

You need to add the appropriate JCL to the CICS processing JCL to assign a tape unit to the system number, SYS030. The following example is for a tape with 80-character records blocked 10 records per block. The tape is unloaded when reading completes.

```

DFHDCT TYPE=EXTRA,           C
      DESTID=yyyy,           C
      DSCNAME=TAPEIN
DFHDCT TYPE=SDSCI,           C

```

DSCNAME=TAPEIN,	C
BLKSIZE=800,	C
RECFORM=FIXBLK,	C
RECSIZE=80,	C
REWIND=UNLOAD,	C
TYPEFLE=INPUT,	C
DEVICE=TAPE,	C
DEVADDR=SYS030,	C
FILABL=NO	

For more information about defining terminal control table and destination control table entries, refer to the appropriate CICS resource definitions guide for your environment.

Considerations for Using DB2/VSE in VSE CICS

This section describes connection considerations, and discusses recovery and database integrity issues for DB2/VSE programs running in the VSE CICS environment.

For information on designing and generating DB2/VSE programs for the VSE CICS environment, refer to the *VisualAge Generator Design Guide* or the *VisualAge Generator Generation Guide*.

For information on preparing DB2/VSE programs for running in the VSE CICS environment, see Chapter 35. Preparing and Running Generated Programs in VSE CICS.

Connection Considerations

For DB2/VSE programs running on VSE CICS, a CONNECT statement is not required to establish a connection between DB2/VSE and your program. To explicitly connect the user ID to the database, you can specify the user ID and password in a CALL EZECONCT service routine in your program. Otherwise, the VSE CICS support for DB2/VSE issues an implicit CONNECT to establish the database connection at the first SQL statement. If you signed on as a VSE/ESA IUI (interactive user interface) user, the implicit CICS user ID is the one connected with your VSE/ESA user ID. Otherwise, it is the CICS default set by the CIRB transaction that starts and stops DB2/VSE online support.

A generated DB2/VSE program running on VSE CICS cannot access multiple databases or switch to a different database. Only one database can be active in any one CICS partition.

Recovery and Database Integrity Considerations

VisualAge Generator programs can use all the recovery and data integrity features that are provided by SQL in the VSE CICS environment.

Relational databases are recoverable resources.

VisualAge Generator programs can use all the recovery and data integrity features that are provided by DB2 in the MVS CICS environment or SQL in the VSE CICS environment.

Relational databases are recoverable resources. If your program makes changes to a relational database, the changes are not committed to the database until the end of a logical unit of work (LUW). If your program ends abnormally before the end of an LUW, all changes that were made since the beginning of the LUW are backed out. See “Recovery and Database Integrity Considerations” on page 311 for additional information.

Considerations for Using DL/I in VSE CICS

This section discusses DL/I database support and recovery and database integrity considerations for DL/I programs running in the VSE CICS environment.

For information on designing and generating DL/I programs for the VSE CICS environment, refer to the *VisualAge Generator Design Guide* or the *VisualAge Generator Generation Guide*.

For information on preparing DL/I programs for running in the VSE CICS environment, see Chapter 35. Preparing and Running Generated Programs in VSE CICS.

DL/I Database Support

The DL/I DOS/VS application control table (ACT) defines which CICS programs are authorized to schedule which PSBs. You must include an entry in the ACT for each generated program that accesses DL/I databases. The program name specified in the ACT entry is the phase name for the program, the same name specified in the CICS processing program table (PPT) entry.

The following example shows a sample ACT that includes an entry for program EX10AD, in which it schedules PSBs EX99PS1 and EX99PS2.

```
// JOB DLZACT02
// OPTION CATAL,NODECK,DUMP
// LIBDEF *,SEARCH=PRD2E.ELACOMM
```

```

// LIBDEF PHASE,CATALOG=PRD2E.ELACOMM
// EXEC ASSEMBLY
XXXX  DLZACT TYPE=INITIAL,SUFFIX=02
      DLZACT TYPE=CONFIG,MAXTASK=3,PI=NO,BFRPOOL=1
      DLZACT TYPE=PROGRAM,PGMNAME=EX10AD,           C
          PSBNAME=(EX99PS1,EX99PS2)
      DLZACT TYPE=BUFFER,HDBFR=(8)
      DLZACT TYPE=FINAL
      END

/*
  ENTRY DLZNUC
// EXEC LNKEDT
/*
/&

```

Refer to the *DL/I DOS/VSE Resource Definition and Utilities* manual for more detailed information on defining and generating an ACT table for DL/I programs.

Specify the DLI keyword in the CICS system initialization table (SIT) to indicate which ACT is to be used with the CICS region:

DLI=YES

Causes the DLZNUC module to be loaded at CICS initialization

DLI=xx

Causes the DLZNUCxx module to be loaded, where xx is the two-character code specified in the SUFFIX keyword used in defining an ACT entry

Recovery and Database Integrity Considerations

VisualAge Generator programs can make use of all the recovery and data integrity features that are provided by DL/I in the MVS CICS and VSE CICS environments.

DL/I databases are recoverable resources. If your program makes changes to a DL/I database, the changes are not committed to the database until the end of a logical unit of work (LUW). If your program ends abnormally before the end of an LUW, all changes that were made since the beginning of the LUW are backed out. See “Recovery and Database Integrity Considerations” on page 311 for additional information.

Chapter 32. System Considerations for VSE Batch

This chapter describes additional system requirements and considerations for administering VisualAge Generator Server in the VSE batch environment.

The following information is discussed:

- Using and allocating VSAM data files
- Using sequential data files
- Determining when database changes are committed
- Considerations for using DB2/VSE
- Considerations for using DL/I
- Runtime JCL

Using and Allocating VSAM Data Files in VSE Batch

Before a VisualAge Generator program can use a VSAM data file in VSE batch for input, you must define and load the data set. sample JCL for defining and loading VSAM data sets.

Using Sequential Data Files in VSE Batch

In the VSE batch environment, each file with a file type of SEQ is associated with a system logical unit number at generation time either by default or by a specific resource association. Resource association is accomplished by using the SYSNUM parameter of the ASSOCIATE command. The system logical unit number appears on the SELECT statement in the generated COBOL program. The system logical unit also is used in the generated runtime JCL. The system logical unit is assigned to the physical device where the sequential file resides or will reside. You can assign the system logical unit to SYSLST for an output SEQ file or to SYSIPT for an input SEQ file.

Consider the following information when customizing the sample runtime JCL generated for SEQ files:

- In all cases (disk, tape, or VSE/POWER LST files), an ASSGN statement is required in the JCL to assign the system logical unit to a physical device. Multiple logical units can be assigned to one physical device within the same partition. Additionally, multiple SEQ files can exist on the same device, all using the same SYSNUM value. In this case, only one ASSGN statement is necessary in the JCL for these SEQ files.

The following sample JCL shows what your JCL would look like if multiple logical units are assigned to one physical device:

```
// DLBL SEQIN, 'USER.FILE1' , ,SD,BLKSIZE=80
// EXTENT SYS005,volid1,1,0
// ASSGN SYS005,ANYDISK,VOL=volid1,SHR
// DLBL SEQOUT, 'USER.FILE2' , ,SD,BLKSIZE=88
// EXTENT SYS005,volid1,1,0,302,100
```

More typically, different system logical units are used for files on different physical volumes, as shown in the sample JCL that follows:

```
// DLBL SEQIN, 'USER.FILE1' , ,SD,BLKSIZE=80
// EXTENT SYS005,volid1,1,0
// ASSGN SYS005,ANYDISK,VOL=volid1,SHR
// DLBL SEQOUT, 'USER.FILE2' , ,SD,BLKSIZE=88
// EXTENT SYS006,volid2,1,0,302,100
// ASSGN SYS006,ANYDISK,VOL=volid2,SHR
```

Sample JCL is provided in the output appl-name JCX file at generation time, and can be customized for use with your sequential (SEQ) files. Refer to the *VSE/ESA System Control Statements* manual for more information on assigning logical unit device numbers to physical devices using the ASSGN statement.

- For SEQ files on disk, a DLBL statement is needed. Unless the SEQ file is a sequential (SAM) file in VSAM-managed space, the BLKSIZE parameter must be coded on the DLBL statement. BLKSIZE should be specified as a multiple of the record size for an input file and as a multiple of the record size plus eight for an output file. An EXTENT statement is required for each sequential (SAM) file not residing in VSAM-managed space.
- For SEQ files on tape, a TLBL statement is required.
- For a LST file, the system logical unit is assigned to the system logical unit SYSLST. A JECL statement defines the attributes of list output as spooled by VSE/POWER. An example of the JECL statement follows:
* \$\$ LST LST=SYSnnn
- For SEQ files used for the EZEPRINT data set, the file name is truncated because the file name for a DLBL or TLBL statement is limited to seven characters. If the SEQ file is directed to tape, the TLBL file name is EZEPRIN. If the SEQ file is directed to disk, the DLBL file name is EZEPRIN.

Note: A warning message is received during the COBOL compilation when SEQ files are used for the EZEPRINT data set in VSE batch.

- For a SEQ input file assigned to the system logical unit SYSIPT, input records must be provided within the SYSIPT area. These records must be preceded by two semi-colons (;) on a separate line. The SYSIPT area is also used to input the DB2/VSE user ID and password (SID) parameter,

optionally, database server identifier (SDB) parameter, and trace parameters if required. These items must be ordered in SYSIPT as follows:

1. SID and SDB parameters
2. Trace parameters
3. ;;
4. Data records

For example, if you have data records only in the SYSIPT area, your JCL might look like this:

```
// EXEC PGM=PROG1
;;
Record 1
Record 2
/*
```

If you have trace parameters and data records in the SYSIPT area, your JCL might look like this:

```
// EXEC PGM=PROG1
Trace parameters
;;
Record 1
Record 2
/*
```

Determining When Database Changes are Committed

This section presents information about database integrity issues for programs running in the VSE batch environment.

By understanding the concept of a logical unit of work (LUW) in VisualAge Generator programs, you are able to determine when changes are committed to your database. This section presents a definition of an LUW that can aid you in designing your recovery procedures.

An LUW ends whenever a commit point or a rollback occurs.

A commit point occurs when one of the following occurs:

- The top-level program in a run unit ends.
For VSE batch, a run unit consists of all VisualAge Generator programs and non-VisualAge Generator programs that transfer control among themselves. VisualAge Generator programs can use a CALL statement or a DXFR statement to transfer control among other VisualAge Generator programs.
- A program calls the EZECOMIT or COMMIT service.

For VSE batch, VisualAge Generator programs that do not use DL/I issue a commit point only if the program has made changes to an SQL table. A commit point does not occur for changes to an SQL table made by a non-VisualAge Generator program.

A rollback occurs when one of the following occurs:

- A program calls the EZEROLLB or RESET service routine.
- A program ends because of an error condition.

Rollback functions are supported only for DB2/VSE databases in VSE batch. If you want to rollback changes to DL/I databases, you must run the DL/I DOS/VS backout utility if the batch job is not successful.

When a rollback occurs, all changes that were made to a database and all recoverable files since the start of the LUW are backed out.

Considerations for Using DB2/VSE in VSE Batch

This section discusses the following considerations for programs using DB2/VSE in VSE batch:

- Connection considerations for SQL programs
- Starting SQL programs in VSE batch
- Recovery and database integrity considerations

For information on designing and generating SQL/DS programs for the VSE environment, refer to the *VisualAge Generator Design Guide* and the *VisualAge Generator Generation Guide*.

For information on preparing DB2/VSE programs for running in the VSE environment, see “Preparing to Run a DB2/VSE Program” on page 346.

Connection Considerations

In VSE batch, the DB2/VSE CONNECT statement is required to establish a connection between DB2/VSE and your program. A generated DB2/VSE program can establish the database connection by one of the following methods:

- Calling the EZECONCT service routine to connect the specific user ID to a database server.

This call must be issued before any other SQL request in the run unit.

- Allowing VisualAge Generator Server to use the user ID, password, and the optional database server name to connect the user to a database server.

VisualAge Generator Server uses the values you specify in the SID and SDB parameters in the system logical unit SYSIPT in the VSE batch JCL. If the SDB parameter is not specified, the connection is made to a default database server defined in the DBNAME directory (ARISDIRD.A) cataloged in the DB2/VSE production library.

The format of the SID and SDB parameters is shown below:

```
SID=uuuuuuuu/pppppppp, SDB=ssssssssssssssss;
```

or

```
SID=uuuuuuuu/pppppppp SDB=ssssssssssssssss;
```

or

```
SID=uuuuuuuu/pppppppp
```

where:

uuuuuuuu

Is the DB2/VSE user ID

This can be from 1 to 8 characters.

pppppppp

Is the DB2/VSE user password

This can be from 1 to 8 characters.

ssssssssssssssss

Is the database server identifier (optional)

This can be from 1 to 18 characters. If the SDB parameter is not specified, the connection is made to a default database server.

The following sample JCL shows how to use SYSIPT input for the DB2/VSE user ID and password, and the database server identifier.

```
// JOB SQL BATCH PROCESS ( Application with DB2/VSE)
// ASSGN SYS004, SYSIPT
// LIBDEF PHASE, SEARCH=(PRD2E.ELACOMM, PRD2E.SQL340, PRD5.TENGR22U)
// EXEC PGM=BATSQL1, SIZE=AUTO
SID=SQLDBA/SQLDBAPW SDB=SQL_340;
/*
/ &
```

A generated DB2/VSE program running under VSE batch can access multiple DB2/VSE program servers. A generated DB2/VSE program can switch to a different database by calling the EZECONCT service routine. However, you must insure that the program user has database CONNECT authority for this function.

Considerations for Processing DB2/VSE Programs

The way you start a generated DB2/VSE program in batch depends on whether you run in multiple- or single-user mode.

Use multiple-user mode if the relational database is to be shared among CICS users, or batch jobs, or both. For multiple-user mode, use the same JCL that you would for an ordinary batch job. The system operator must have started DB2/VSE before your job runs in multiple-user mode.

Use single-user mode if no other CICS transactions or batch jobs need to obtain access to the relational database while your job runs. In single-user mode, you start up DB2/VSE and pass it the name of the VisualAge Generator Server initialization program. DB2/VSE passes control to VisualAge Generator Server to start your program.

Recovery and Database Integrity Considerations

VisualAge Generator programs can use all the recovery and data integrity features that are provided by DB2/VSE.

Relational databases are recoverable resources. If your program makes changes to a relational database, the changes are not committed to the database until the end of an LUW. If your program ends abnormally before the end of an LUW, all changes that were made since the beginning of the LUW are backed out. See “Determining When Database Changes are Committed” on page 327 for information describing when an LUW ends.

Considerations for Using DL/I in VSE Batch

This section contains considerations for using DL/I, such as DL/I database support, checkpoints, and database recovery considerations.

For information on designing and generating DL/I programs for the VSE environment, refer to the *VisualAge Generator Design Guide* and the *VisualAge Generator Generation Guide*.

For information on preparing DL/I programs for running in the VSE environment, see “Chapter 13. Preparing Generated Programs on MVS Systems” on page 123.

DL/I Support in VSE Batch Programs

In programs generated for the VSE batch environment, you can obtain access to either DL/I databases, DB2/VSE databases, or both. If access is required to

both DL/I and DB2/VSE databases in the same program, it is the program developer's responsibility to maintain database integrity because DL/I and DB2/VSE changes are committed independently.

When a generated DL/I program is run, it actually runs under DL/I control, which can either be batch or MPS (multiple partition support) batch. In DL/I MPS batch, the restart facility can be specified for database recovery and restart if a program ends abnormally.

If generated batch programs gain access to online DL/I databases as MPS batch jobs, you must also include an entry for the MPS batch partition controller program (DLZBPC00) in the DOS/VS DL/I program control table (ACT). Include the PSB name for each PSB used with an MPS batch job on the program entry for DLZBPC00. See "DL/I Database Support" on page 323 for an example of how to code an ACT entry.

Recovery and Database Integrity Considerations

In a DL/I program running in VSE batch, a commit point results in a DL/I CHKP (checkpoint) call. The contents of the EZEDLPSB special function word are used as the checkpoint identifier. After the CHKP call, the EZEDLSTC special function word contains the status code that is returned. A call to the EZEROLLB service routine is ignored because the DL/I ROLLBACK function is not supported in VSE batch.

See "Determining When Database Changes are Committed" on page 327 for information describing when a commit point occurs. If you want to rollback changes to DL/I databases, you must run the DL/I DOS/VS backout utility if the batch job is not successful.

Performance Considerations for VSE Batch

See "Phases in Memory" on page 292 for information on performance considerations and the methods used to place phases in memory. These methods are particularly beneficial if the VisualAge Generator program is being called repeatedly by a non-VisualAge Generator program.

Runtime JCL

See "Chapter 36. Preparing and Running Generated Programs in VSE Batch" on page 357 for examples of batch runtime JCL.

Part 7. Preparing and Running Generated Applications on VSE Systems

Chapter 33. Output of Program Generation on VSE Systems	335
List of Program Preparation Steps after Program Generation.	335
Outputs of Generation	335
Objects Generated for Programs.	338
Application COBOL Program	338
Sample Runtime JCL	339
Objects Generated for Tables	339
Table COBOL Program	339
Objects Generated for Map Groups.	339
Online Print Services Program	339
Batch Print Services Program.	339
Map Group Format Module	340
Objects Generated for All Part Types (Programs, Tables, and Map Groups)	340
Listing	340
Preparation JCL	340
Preparation Command File	340
CICS Program and Transaction Definitions	341
Parts Control File	341
Chapter 34. Preparing Generated Programs on VSE Systems	343
Allocating Preparation Data Sets	343
Running the Preparation Job	343
Error Processing for Preparation Jobs	343
Analyzing the Preparation Listing	344
Analyzing the DB2/VSE Precompile Step	344
Analyzing the CICS Translator Step	345
Analyzing the Compile Step	345
Analyzing the Link-edit Step	345
Link-edit Errors Related to Missing Modules	345
Other Link-Edit Errors	346
Preparing to Run a DB2/VSE Program	346
Granting Authority to Program Users	346
Defining Synonyms for DB2/VSE Tables	347
Defining a Program Specification Block	347
Chapter 35. Preparing and Running Generated Programs in VSE CICS	349
Modifying CICS Resource Tables	349
Program Entries (PPT)	349
Transaction Entries (PCT)	351
Destination Control Table Entries (DCT)	354
File Control Table Entries (FCT).	354
Using Remote Programs, Transactions, or Files	354
Using VSE/POWER Spool Files	355
Modifying CICS Startup JCL	355
Making New Programs Available in the VSE CICS Environment	355
Making Programs Resident	356
Running Programs under VSE CICS	356
Starting the Transaction in VSE CICS	356
Controlling Diagnostic Information in the CICS Environment	356
Chapter 36. Preparing and Running Generated Programs in VSE Batch	357
Making VisualAge Generator Server, COBOL, and the Generated Programs Available	357
Creating or Modifying Runtime JCL	357
Tailoring JCL before Generation.	358
Modifying Runtime JCL	359
Running Main Programs under VSE Batch	361
Examples of Runtime JCL for VSE Batch Programs	362
Running a Main Batch Program with No Database Access	363
Running a Main Batch Program with DB2/VSE Access.	363
Running a Main Batch Program with DL/I Access	365
Running a Main Batch Program with DB2/VSE and DL/I Access	367
Recovery and Restart for Batch Programs	369
Chapter 37. Moving Prepared Programs to Other Systems from VSE Systems	371
Moving Prepared Programs to Another VSE System	371

Maintaining Backup Copies of Production
Libraries 371

Chapter 33. Output of Program Generation on VSE Systems

This chapter provides an overview of preparing and running a generated program. When generation is complete, you must prepare the programs for production just as you would prepare COBOL programs you might write directly. Preparation steps include running translators, precompilers and compilers, performing link-edits, and defining control tables for the target runtime environment.

List of Program Preparation Steps after Program Generation

VisualAge Generator Server supports program preparation and installation in the VSE environments using the generated preparation JCL and cataloged procedures shipped with VisualAge Generator Server. You must perform the steps listed in Table 48 before you can run your program in a VSE target environment.

Table 48. Preparation Steps for VSE Environments

Preparation Step	Environment
DB2/VSE precompile	DB2/VSE programs only
CICS translation	CICS only
COBOL compile	All
Link	All
CICS program and transaction definition	CICS

The process of generating a program for VSE environments is described in the *VisualAge Generator Generation Guide*.

Outputs of Generation

When you generate programs, map groups, or tables for the VSE environment using VisualAge Generator Developer on the workstation, several types of objects are produced. Most of these objects must be copied to a VSE system and processed further before the program is ready to run. At your request, the generator produces a workstation command file to copy the generated objects to your VSE system and submits a VSE job to complete preparation processing for the generated objects. Once preparation is complete, objects not required for running the program are deleted from the VSE library.

Table 49 provides information about the types of VSE objects produced by generation, including:

- Type of object produced
- OS/2 file name and extension
- VSE library member name and type
- Runtime environments for which the object is produced (VSE CICS or VSE batch)
- Generation options that control whether the object is produced
- Whether the member is deleted from the VSE library after preparation
- Whether the object is modifiable

Refer to the *VisualAge Generator Generation Guide* for more information on controlling and modifying generation and preparation of VSE objects.

Descriptions of the objects begin on page 338.

Table 49. Objects Produced for Programs by the Generation Facility for VSE

Object Type	OS/2 File Name and Extension	VSE Library Member Name	VSE Library Member Type	Runtime Environment	Generation Option	Keep after Preparation	Modifiable Object
Application COBOL program	applname.CBL	applname	C	VSE CICS and VSE batch	None	No	No
Sample Runtime JCL	applname.JCX	applname	JCX	VSE batch	/RUNFILE	Yes	Review and possible modification required

Table 50. Objects Produced for Tables by the Generation Facility for VSE

Object Type	OS/2 File Name and Extension	VSE Library Member Name	VSE Library Member Type	Runtime Environment	Generation Option	Keep after Preparation	Modifiable Object
Table COBOL program	tablename.CBL	tablename	C	VSE CICS and VSE batch	/GENTABLES	No	No

Table 51. Objects Produced for Map Groups by the Generation Facility for VSE

Object Type	OS/2 File Name and Extension	VSE Library Member Name	VSE Library Member Type	Runtime Environment	Generation Option	Keep after Preparation	Modifiable Object
Online print services COBOL program - See note 1	mapgname.CBL	mapgname	C	VSE CICS	/GENMAPS and /GENHELPMAPS	No	No
Batch print services COBOL program - See note 1	mapgnameP1.CBL	mapgnameP1	C	VSE batch	/GENMAPS and /GENHELPMAPS	No	No
Map group format module - See note 2	mapgnameFM.FMT	mapgnameFM	OBJ	VSE CICS	/GENMAPS and /GENHELPMAPS	No	No

Table 52. Objects Produced for All Part Types by the Generation Facility for VSE (Programs, Tables, and Map Groups)

Object Type	OS/2 File Name and Extension	VSE Library Member Name	VSE Library Member Type	Runtime Environment	Generation Option	Keep after Preparation	Modifiable Object
Listing	mbrname.LST	Not copied to host		VSE CICS and VSE batch	/LISTING, /LISTING, and ONERROR	No	No
Control Parts File - See note 3	mbrname.PCF	PCF		VSE CICS and VSE batch	None	No	No
Preparation Command File	mbrname.PRP	Not copied to host		VSE CICS and VSE batch	/PREPFILE	No	Yes
Preparation JCL	mbrname.JCP	mbrname	Z	VSE CICS and VSE batch	/PREPFILE	No	Yes

Table 52. Objects Produced for All Part Types by the Generation Facility for VSE (Programs, Tables, and Map Groups) (continued)

Object Type	OS/2 File Name and Extension	VSE Library Member Name	VSE Library Member Type	Runtime Environment	Generation Option	Keep after Preparation	Modifiable Object
VSE CICS Program and Transaction Definitions	mbrname.PCT (program MACRO or RDO definitions), mbrname.PPT (transaction MACRO or RDO definitions)	mbrname	PCT and PPT	VSE CICS	/CICSENTRIES	Yes	Review and possible modification required

Notes:

1. This is produced only if the map group contains print maps.
2. This is produced only if the map group contains terminal maps.
3. If any map groups or tables are generated with a program, a parts control file part is written for each generated table or map group. The part lists the parts generated for the table or map group. The part written for the program lists all parts generated for the program and all parts generated for associated map groups and tables.

Objects Generated for Programs

Application COBOL Program

The generated program is a COBOL program that contains the following:

- Program control logic
- Logic for program processes, statement groups, and I/O operations
- Data for both the program and program control

The program control logic performs the following functions for a program, as needed:

- Initialization
- Cleanup at end of program
- Error reporting
- Segmentation support, including environment saving and restoration
- Transfer

Sample Runtime JCL

The generator produces sample JCL for running programs in the VSE batch environment when the /RUNFILE generation option is specified during program generation. The JCL does not contain a JOB statement. Each person using the JCL must provide a JOB statement.

The JCL is produced from model JCL templates that can be modified to enforce customer data-set naming conventions. Refer to the *VisualAge Generator Generation Guide* for information about modifying the sample templates.

The JCL might not be complete and should be reviewed and modified if necessary before being used. For example, the JCL for the generated program does not contain any DLBL statements for data sets used by other programs that can be started by a CALL or DXFR statement. Comments in the JCL indicate where the DLBL statements for these programs need to be added. To build the final JCL needed to run a set of programs as a run unit, you should edit the program JCL and include the DLBL statements for invoked programs with the JCL for the first main program. You might need to add DLBL statements for files that are used during run time with the EZEDEST or EZEDESTP special function words.

Objects Generated for Tables

Table COBOL Program

The table program is a COBOL program that contains the table contents defined in program working storage. This object is produced when you specify the /GENTABLES generation option. This allows tables to be generated independently of programs when the contents of a table need to be changed.

Objects Generated for Map Groups

Online Print Services Program

The online print services program is a COBOL program that performs printer I/O and SET map CLEAR and EMPTY functions for printer maps for VSE CICS programs. This object is produced when you specify the /GENMAPS or the /GENHELPMAPS generation options.

Batch Print Services Program

The batch print services program is a COBOL program that performs printer I/O and SET map CLEAR and EMPTY functions for printer maps for VSE

batch programs. This object is produced when you specify the /GENMAPS or the /GENHELPMAPS generation options.

Map Group Format Module

The map group format module is a generated structure that describes the map layout for terminal maps in the map group. The generator builds the structure as a VSE object module for the VSE CICS environment. This object is produced when you specify the /GENMAPS or the /GENHELPMAPS generation options. A help map group object is generated with the program if you specify the /GENHELPMAPS generation option. The preparation job stores the map group format module as a phase with the same name during the link-edit step.

Objects Generated for All Part Types (Programs, Tables, and Map Groups)

Listing

A source listing of the part being generated and its associates is produced if one of the print options (/LISTING or /LISTINGONERROR) is specified during generation. The listing includes any messages that were issued and options that were in effect during generation.

Preparation JCL

The generator produces JCL statements to precompile, translate, compile, and link generated programs. The JCL produced depends on the nature of the program:

- DB2/VSE requires a precompile step.
- VSE CICS programs require CICS translation and must link with VSE CICS modules.
- A program that issues DB2/VSE calls must link with the appropriate DB2/VSE modules.
- A program that issues DL/I calls must link with the appropriate DL/I modules.
- Map group format modules require a link-edit step.
- COBOL programs must compile and link.

The JCL is produced from model JCL templates that you can modify to enforce file-naming conventions or preparation procedures.

Preparation Command File

A preparation command file is generated when the /PREPFILE generation option is specified. The preparation command file produced by a GENERATE

subcommand contains logic to prepare the generated parts. If the generated program includes tables and maps, the logic for preparing them is contained in the preparation command file.

The preparation command file runs on the workstation and contains commands used to transfer the source files and preparation JCL file to the target host system. One preparation command file is created for each GENERATE subcommand that is issued. Refer to the *VisualAge Generator Generation Guide* for more information.

CICS Program and Transaction Definitions

CICS program and transaction definitions for the VSE CICS environment are generated as specified in the /CICSENTRIES option. The definitions might need to be modified before using in the CICS system definition.

Parts Control File

The parts control file is a list of all objects produced by the generator the last time generation was performed for the program, table, or map group with the same name as the parts control file part. This list provides a record of generated objects.

Chapter 34. Preparing Generated Programs on VSE Systems

This chapter describes preparation procedures that are common to all runtime environments. These preparation procedures include the following steps:

- Running the preparation job
- Analyzing the preparation listing
- Preparing and running a DB2/VSE program
- Defining a PSB for a DL/I program

Allocating Preparation Data Sets

A job should be run during the VisualAge Generator Server installation process to create the PRD2.EZELIB data set on the VSE system. All generated programs are stored in PRD2.EZELIB by default. The data set can be changed for any program during program generation, but you must specify a data set that has already been created.

Running the Preparation Job

The preparation function of VisualAge Generator submits the preparation JCL file built by the generator to the VSE/POWER reader. If the preparation job ends with an error that can be corrected without requiring the program to be generated again, you can use the VisualAge Generator PREPARE subcommand to submit the preparation JCL file to the VSE/POWER reader again. You need to use your own JOB statement with the file.

Error Processing for Preparation Jobs

Depending on the type of program being generated, VisualAge Generator Server preparation jobs can contain multiple VSE jobs (DB2/VSE preprocessor, CICS translator, COBOL compiler, VSE Librarian, and VSE linkage editor). Because each job is dependent on the successful completion of the previous job, it wastes system resources to run jobs when the preceding job has an error condition. To help you avoid unnecessary work, VSE preparation jobs uses error condition checking.

At the end of each job, a variable is set using the SETPARM command in a VSE procedure indicating either a “good” return code or a “bad” return code. This procedure is then run at the beginning of each of the remaining jobs to determine whether the prior job was successful or not. If a “bad” return code

is detected, this job and all remaining jobs in the preparation jobs are cancelled. If a “good” return code is detected, the current job runs to completion, then sets the return code in the same procedure depending on whether the job ran successfully or not. The process continues for all remaining jobs that are included in the preparation jobs.

The procedure that contains the SETPARM variable is created with a unique name each time the preparation jobs is run. You will see the following message on the VSE console while the first job in the preparation jobs is running:

```
IESI0230I MEMBER XXX.PROC DOES NOT EXIST IN LIBRARY SSSS.LLLLLL  
PROCESSING CONTINUES, IN CASE OF NO  
IESI0230I(+) ERRORS OCCUR THE MEMBER WILL BE CREATED.
```

where XXX is the unique procedure name used to store the SETPARM variable and SSSS.LLLLLL is the VSE/ESA sublibrary specified by the /SYMPARM='PROCLIB' option in VisualAge Generator Developer generation.

The last job in the preparation package is a CLEANUP job. The purpose of this job is to delete the return code procedure that was used in the previous jobs. After deleting the return code procedure, the job issues a GOTO \$EOJ command and immediately end processing. Once the job is complete, you will see the message:

```
1S78I JOB TERMINATED DUE TO 'GOTO $EOJ'
```

on the VSE/ESA console.

Analyzing the Preparation Listing

If you have errors in the generated preparation JCL, check your generation listing for error messages related to the generation of JCL. Errors can occur if the JCL templates used to create the preparation JCL are not modified correctly. Refer to the *VisualAge Generator Generation Guide* for information about modifying the templates.

For error codes you might encounter during the preparation steps, see Chapter 43. Common Messages during Preparation for VSE Systems.

Analyzing the DB2/VSE Precompile Step

If you receive an error message from the DB2/VSE precompiler, see “Common DB2/VSE Return Codes” on page 445. If you cannot determine the cause of the error from this manual or from the appropriate DB2/VSE messages manual, have your system administrator contact the IBM Support Center.

The precompile step is successful if the return code is 4 or less.

Analyzing the CICS Translator Step

You should not receive any messages from the CICS translator. If you receive messages and no errors are found in the generator step, have your system administrator contact the IBM Support Center.

The translator step is successful if the return code is 4 or less.

Analyzing the Compile Step

Several common compiler warning messages are possible if you do any of the following:

- Use SQL row records
- Use the OPTIMIZE compile option
- Assign or move a data item to itself
- Use /MATH=COBOL
- Display printer maps in VSE Batch

These common messages are listed in “COBOL Compilation Messages” on page 429 . If you encounter COBOL messages other than those listed in that section, check the listing file from generation for any error messages. Correct these errors and then generate and prepare the program again. If there are no errors in the listing file, give the COBOL compile listing that contains the error and the associated COBOL code to your system administrator and have the administrator contact the IBM Support Center.

The compile step is successful if the return code is 4 or less.

Analyzing the Link-edit Step

This section discusses how to resolve link-edit errors.

Link-edit Errors Related to Missing Modules

Some modules from products such as CICS, DB2/VSE, and VisualAge Generator Server must be included in the phase. A small set of VisualAge Generator Server functions are linked with each generated COBOL program. The VisualAge Generator Server functions are included with one or more linkage editor statements in the following form:

```
INCLUDE modulename
```

If you generated your program to use dynamic linkage for calls and transfers to other programs, then the link-edit step should always end with a return code of 0. If you receive a nonzero return code, review your link-edit listing for "module not found" messages. Check that the link-edit step includes a LIBDEF statement that searches the sublibrary that contains the missing module.

The link-edit step may end with a nonzero return code if you generated your program to use calls to another program and have defined a linkage editor control statement file (file name member.LED) for generation. The linkage editor control statement file is used by the generator as link-edit input for linking the program. The link-edit step ends with a return code of 8 until all the programs to be linked together have been generated and prepared. If all the programs have been prepared, the link step should give a return code of 0, indicating that the program is ready to run. Refer to the *VisualAge Generator Generation Guide* for additional information on specifying the linkage editor control statements for statically linked modules.

Other Link-Edit Errors

If you receive a nonzero return code for some reason other than a module not found in the initial link-edit step, refer to the VSE/ESA messages and codes manual for an explanation of the messages. If the link-edit step is not successful using the linkage editor control statements you provided, review the information about specifying linkage editor control statements for static linkage in the *VisualAge Generator Generation Guide*, and refer to the error information in the VSE/ESA messages and codes manual to determine the cause of the error.

Preparing to Run a DB2/VSE Program

This section describes the tasks that need to be done to enable program users to run a DB2/VSE program, such as granting authority and defining synonyms for DB2/VSE tables.

Granting Authority to Program Users

When you have finished preparing your generated program, you need to grant authority to the program users so they can run the program using DB2/VSE. You can do this by using the following GRANT command:

```
GRANT RUN ON myapp1 TO user1, user2, user3
```

If the KEEP parameter (the default) is specified in the DB2/VSE precompile for a program, you need to grant authority one time for each program user. If the REVOKE parameter is specified, you must grant authority each time the program is precompiled.

Defining Synonyms for DB2/VSE Tables

Depending on how you specified the table name when you defined your SQL row records in the program, you might need to create a synonym for the table before you can run the program. If you do not fully qualify a table name in an SQL row record with the table creator identifier, the database manager supplies a default qualifier.

For static SQL statements within your program, the default qualifier is the DB2/VSE authorization ID that is used during program preparation. Refer to the appropriate DB2/VSE manual for more information. If you are using dynamic SQL statements within your program, the default qualifier is the runtime authorization ID. All VisualAge Generator programs run in static mode. If you use table name host variables or runtime statement build, individual SQL statements use dynamic mode.

If DB2/VSE synonyms are used for table names, the program mode determines who must create the synonyms. If the SQL row record is used with dynamic SQL statements in a generated program, each program user must create the synonym for the table. If the SQL row record is used only in static SQL statements in a generated program, only the programmer identified in the precompile operation needs to create a synonym for the table. If you specified a fully qualified table name when you defined your SQL row record, then you do not need to do anything special when you precompile the program.

For additional information on using unqualified table names and synonyms, refer to the *VisualAge Generator Design Guide*.

Defining a Program Specification Block

Define a DL/I DOS/VS program specification block (PSB) that matches the PSB specified for the generated program, omitting any TP and GSAM PSBs included in the program PSB.

Chapter 35. Preparing and Running Generated Programs in VSE CICS

This chapter describes the unique steps required to prepare and run a generated COBOL program in a VSE CICS environment:

- Modifying CICS resource tables
- Modifying CICS startup JCL
- Making new programs available
- Making programs resident
- Running programs

For general information on preparing your program for the runtime environment, see Chapter 34. Preparing Generated Programs on VSE Systems.

Modifying CICS Resource Tables

The VSE CICS environment uses tables to identify startup parameters, transactions, programs, files, databases, transient data destinations, and system locations for proper operation. You must add to or modify these tables to correctly identify all objects to be used in the new or changed program. The CICS tables are compiled as assembler programs and stored in a library. Some tables can also be maintained through an online facility as described in the resource definition online manual for your version of CICS.

The following examples show sample entries for the various CICS table types needed for runtime support of generated programs. Refer to the CICS operations guide for instructions on assembling and link-editing CICS tables. Refer to the section on developing client/server programs in the *VisualAge Generator Design Guide* for special considerations related to remote files and client/server.

Program Entries (PPT)

The entries in the processing program table (PPT) define programs to CICS. The VisualAge Generator generation process creates programs that must be defined in the PPT or as a resource definition online (RDO) PROGRAM entry.

An entry is required for each VisualAge Generator table, main program, dynamically called program, map group format module, and print services program. When you generate a program, map group, or table, you can determine what programs were generated by looking at the parts control file

for the generated program, map group, or table. Each entry with a part type of APPL, MAPGROUP, MAPFM, or TABLE requires a corresponding entry in the PPT.

If you specify /CICSENTRIES=MACRO during generation, PPT entries are generated for you from templates for each program that requires a PPT entry. The preparation command file created by generation copies the table entries to the VSE library specified at generation. The library parts have the part name equal to the name of the generated object (program, map group, or table) and the part type of PPT. Refer to the *VisualAge Generator Generation Guide* for information on how to modify templates for generating PPT entries.

The following figures show sample PPT entries generated from the default templates shipped with VisualAge Generator.

```

DFHPPT TYPE=ENTRY,           X
      PROGRAM=applname,      X
      PGMLANG=COBOL,         X
      PGMSTAT=ENABLED,       X
      RELOAD=NO,             X
      RES=NO                  1

```

Figure 56. PPT Entry for Generated Program

1 - RES=YES might provide better performance for frequently used programs.

```

DFHPPT TYPE=ENTRY,           X
      PROGRAM=tblname,       X
      PGMLANG=COBOL,         X
      PGMSTAT=ENABLED,       X
      RELOAD=NO,             X
      RES=NO                  1

```

Figure 57. PPT Entry for Generated Table

1 - RES=YES might provide better performance for frequently used single user tables.

```

DFHPPT TYPE=ENTRY,           X
      PROGRAM=mapgrp,        X
      PGMLANG=COBOL,         X
      PGMSTAT=ENABLED,       X
      RELOAD=NO,             X
      RES=NO                  1

```

Figure 58. PPT Entry for Online Print Services Program

1 - RES=YES might provide better performance for frequently used map groups.

DFHPPT TYPE=ENTRY,	X	
PROGRAM=mapgrpFM,	X	
PGMLANG=ASSEMBLER,	X	1
PGMSTAT=ENABLED,	X	
RELOAD=NO,	X	
RES=NO		2

Figure 59. PPT Entry for Map Group Format Module

- 1 ASSEMBLER is required for the FM module.
- 2 RES=YES might provide better performance for frequently used map groups.

If you specify /CICSENTRIES=RDO, CICS RDO DEFINE PROGRAM commands are generated for you from templates for each program that requires an RDO PROGRAM entry. The preparation command file created during generation copies the RDO command files to the VSE library specified at generation. Refer to the *VisualAge Generator Generation Guide* for information on how to modify templates for generating RDO commands.

The following examples show RDO commands generated from the default templates shipped with VisualAge Generator.

- The DEFINE PROGRAM command for a COBOL program:
CEDA DEF PROG(progname) L(COBOL) REL(NO) RES(NO) S(ENABLED)
- The DEFINE PROGRAM command for a map group format module:
CEDA DEF PROG(mapgrpFM) L(ASSEMBLER) REL(NO) RES(NO) S(ENABLED)

The values shown for REL, RES, and S keywords are the default values and can be omitted from the command. RES(YES) might provide better performance for frequently used programs.

Transaction Entries (PCT)

The PCT contains the control information used by CICS for identifying and initializing a transaction. This table is required by CICS to verify incoming requests to start transactions, and to supply information about the transaction such as the transaction priority, the security key, and the length of the transaction work area (TWA).

A CICS PCT entry or an RDO TRANSACTION entry is required for each transaction code used to start a VisualAge Generator-generated main program. This includes transactions with the following attributes:

- Started from a clear CICS screen
- Used as an EZESEGTR name
- Used as the target of an XFER or CALL CREATX statement
- Transferred to by a non-VisualAge Generator program
- Started by other CICS facilities
- Used as the name of the segmented transaction ID specified at generation

Programs that are invoked through a DXFR or CALL statement do not require a PCT entry.

If you specify /CICSENTRIES=MACRO, CICS PCT entries are generated for you from templates for each main batch or main transaction program. Primary and restart transaction identifiers are specified using the /TRANSID generation option. The default is a single transaction definition using the first 4 characters of the program name as the transaction name. The preparation command file created during generation copies the table entries to the VSE library specified at generation. The library parts have the part name equal to the name of the program with a part type of PCT. Refer to the *VisualAge Generator Generation Guide* for information on how to modify templates for generating PCT entries.

The following figures show sample PCT entries generated from the default templates shipped with VisualAge Generator, along with notes describing reasons why you might want to modify the default transaction definitions.

DFHPCT TYPE=ENTRY,	X	
PROGRAM=applname,	X	1
TRANSID=tran,	X	2
DTB=YES,	X	3
DUMP=NO,	X	4
RESTART=NO,	X	5
TWASIZE=1024		6

Figure 60. PCT Entry for Generated Program

- 1** Name of program to be started for this transaction. If the transaction being defined is one that is used as the EZESEGTR value for a segmented CONVERSE process option, specify the name of the program that did the CONVERSE process option for best performance.
- 2** Assign a unique transaction ID.
- 3** Changes that have not been committed are backed out if the program ends abnormally. If the program updates recoverable resources, you must specify DTB=YES. If the program is inquiry-only, specifying DTB=NO might provide better performance. If DTB=YES is specified

on the program control table entries for the transactions, the value specified for DBP in the system initialization table (SIT) is significant. VSE CICS provides two dynamic backout programs, one for systems that require local DL/I support (specify DBP=2\$), the other one for systems that do not require local DL/I support (specify DBP=1\$). The dynamic backout programs are provided by VSE CICS if an entry is included in the PPT that specifies TYPE=GROUP and FN=BACKOUT. Refer to the VSE CICS system definition guide for more information.

- 4 VisualAge Generator Server contains an exit that is activated when your transaction ends abnormally. The exit is enabled by a CICS HANDLE ABEND statement at initialization time. The exit issues CICS DUMP commands if the transaction ends abnormally unless the diagnostic control setting for the transaction is set to bypass dumps. Specifying DUMP=YES causes CICS to issue a transaction dump when transactions end abnormally. If you specify DUMP=YES, use the VisualAge Generator Server diagnostic controller utility to set the dump option to “no dump.” See “Chapter 39. Diagnostic Control Options” on page 383 for more information about the VisualAge Generator Server diagnostic controller utility.
- 5 Specify RESTART=YES if you intend to use the transaction restart facility and the EZEDLRST special word.
- 6 The size for the CICS Transaction Work Area (TWA) must be at least 1024 bytes. If you are running a non-VisualAge Generator program together with VisualAge Generator programs and the program requires a TWA, you must increase the TWASIZE parameter by the amount your non-VisualAge Generator program requires and specify the /TWAOFF option when generating the VisualAge Generator programs. Refer to the *VisualAge Generator Generation Guide* for information about the /TWAOFF generation option.

If you specify /CICSENTRIES=RDO, CICS RDO DEFINE TRANSACTION commands are generated for you from templates for each main batch or main transaction program. Primary and restart transaction identifiers are specified using the /TRANSID generation option. The default is a single transaction definition using the first 4 characters of the program name as the transaction name. The preparation command file created by generation copies the RDO command file into the VSE library specified at generation. The file also contains any DEFINE PROGRAM commands generated for the program. Refer to the *VisualAge Generator Generation Guide* for information on how to modify templates for generating RDO entries.

The following example shows an RDO entry generated from the default templates shipped with VisualAge Generator. The same comments that apply to the previous description of PCT entries also apply to the RDO commands.

```
CEDA DEF TR(tran) PROG(applname) I(BACKOUT) DU(NO) RES(NO) TW(1024)
```

The I(BACKOUT) parameter is equivalent to DTB=YES. DTB=NO cannot be specified with the DEFINE command. If DTB=YES is specified on the program control table entries for the transactions, the value specified for DBP in the system initialization table (SIT) is significant. VSE CICS provides two dynamic backout programs, one for systems that require local DL/I support (specify DBP=2\$), the other one for systems that do not require local DL/I support (specify DBP=1\$). The dynamic backout programs are provided by VSE CICS if an entry is included in the PPT that specifies TYPE=GROUP and FN=BACKOUT. Refer to the VSE CICS system definition guide for more information.

Destination Control Table Entries (DCT)

A CICS destination control table (DCT) entry is required for each program file that is assigned to a transient data queue. A DCT entry is also required for destinations specified as error destination queue names using the VisualAge Generator Server diagnostic controller utility. The parameters for DCT entries depend on your destination type. There are intrapartition, extrapartition, indirect, and remote destinations. Templates for the DCT entries are not provided by VisualAge Generator Developer. See “Using and Allocating Data Files in MVS CICS and VSE CICS” on page 317 for information about defining and managing program data files and “Defining Transient Data Queues” on page 320 for information about defining the DCT entry for the error destination queue. Refer to CICS publications for detailed information about DCT entries.

File Control Table Entries (FCT)

A CICS file control table (FCT) entry is required for each program file that is specified as file type VSAM. You must identify all FCT entries that might be referenced at run time. See “Using and Allocating Data Files in MVS CICS and VSE CICS” on page 317 for more information on defining and managing program data files in the CICS environment.

Using Remote Programs, Transactions, or Files

Refer to the section on developing client/server programs in the *VisualAge Generator Design Guide* for information about the CICS table entries required to support remote programs, transactions, or files. Also refer to the CICS manuals for information about defining remote programs, transactions, or files.

Using VSE/POWER Spool Files

Refer to the *VisualAge Generator Design Guide* for information about using VSE/POWER spool files.

Modifying CICS Startup JCL

The CICS startup JCL might need to be modified to add or change allocations for files used by VisualAge Generator programs. These include VSAM files and extrapartition transient data destinations.

Making New Programs Available in the VSE CICS Environment

The transaction that runs your program must have access either through the permanent library search order or the temporary library search order to your generated program program, tables, and map group format modules as well as called program, VisualAge Generator Server, and LE runtime libraries. Your program must also have access to the libraries that contain DL/I and DB2/VSE if these products are used by the program. Many of these components are in the permanently assigned search order.

After you generate a new version of a program, map group, or table, you need to make the programs available to CICS.

For programs and map groups, you can use the CICS NEWCOPY command or the VisualAge Generator Server new copy utility to cause the new copy of the program to be used the next time a load request is issued for the program. When you specify a part type of Map Group, the VisualAge Generator Server new copy utility makes a new copy for both the online print services program and the map group format module. If you use the CICS NEWCOPY command for a map group, you must issue the NEWCOPY for both the online print services program and the map group format module.

For tables, use the VisualAge Generator Server new copy utility to cause a fresh copy of the table to be used the next time a load request is issued for the table. Do not use the CICS NEWCOPY command for tables. The VisualAge Generator Server new copy utility sets a flag indicating that the new copy of the table is to be used the next time a program loads the table contents.

For more information on the VisualAge Generator Server new copy utility, see “New Copy” on page 377.

Making Programs Resident

You can make frequently used programs or programs with high performance requirements resident to avoid the overhead of loading the programs when they are used. To help you decide which programs should be made resident, you can use CICS shutdown statistics to determine how often a generated program is loaded in a CICS region.

To make application and map group programs resident, specify the program as resident in the PPT or RDO entry for the program. To make table programs resident, specify the table as resident when you define the table.

Running Programs under VSE CICS

Either a main transaction program or main batch program generated for VSE CICS can be started with CICS facilities. Called programs can be started only by another VisualAge Generator program or by a non-VisualAge Generator program.

Prior to starting a generated program, the program user might be required to sign on to the CICS environment. Refer to CICS documentation for information about signing on.

Starting the Transaction in VSE CICS

Any main program or main batch program that is generated with a target environment of VSE CICS can be started by entering the transaction code associated with the main program from a clear screen in CICS. Any main program that is started directly in CICS, by an XFER statement from another program, or by a CREATX service routine must have a unique transaction code assigned to it. The transaction code must be defined in the PCT or with an RDO TRANSACTION entry and be associated with the first program in the run unit.

Controlling Diagnostic Information in the CICS Environment

VisualAge Generator Server provides a diagnostic controller utility for the CICS environment. This utility allows you to control the type of dump, the name of the error destination queue and journal number for error messages, and whether the transaction is disabled when a run-unit error occurs. See “Chapter 39. Diagnostic Control Options” on page 383 for more information about the diagnostic controller utility.

Chapter 36. Preparing and Running Generated Programs in VSE Batch

This chapter describes the unique steps required to prepare a generated program to run in the VSE batch environment:

- Making VisualAge Generator Server, COBOL, and the generated programs available
- Preparing programs for VSE batch
- Creating or modifying runtime JCL
- Running main programs
- Examples of runtime JCL
- Recovery and restart for batch programs

For general information on preparing your program for the runtime environment, see Chapter 34. Preparing Generated Programs on VSE Systems.

Making VisualAge Generator Server, COBOL, and the Generated Programs Available

The job that runs your program must be able to gain access either through the permanent or temporary library search order to your generated program programs, tables, and map format modules as well as called programs, VisualAge Generator Server, and LE runtime libraries. Your program must also be able to gain access to the libraries that contain DL/I and DB2/VSE if these products are used by the program. Many of these components are in the permanently assigned search order.

Creating or Modifying Runtime JCL

This section contains the information you need to modify the runtime JCL created during program generation. If you specify /RUNFILE at generation, the generator creates JCL for running the generated program.

The preparation command file started on the workstation after generation completes copying the runtime JCL file to the specified VSE library. The library member name is the program name. The library member type is JCX. You might need to modify the runtime JCL for one of the following reasons:

- The generator does not include DLBL statements for files required by called or transferred-to programs in the JCL built for the generated program.

- The generator does not include a JOB statement in the JCL.
- The generator does not include a DLBL statement in the JCL for files that are used with the EZEDEST or EZEDESTP special function words.
- The generator does not create any recovery or restart JCL.
- The sample JCL is based on the initial program in the run unit.

The library concatenation must include the VSE sublibraries containing the initial program and any dynamically-invoked programs. This includes program modules that are dynamically called, programs that are transferred-to by DXFR statements, print services programs, map group format modules, and VisualAge Generator table programs.

Tailoring JCL before Generation

VisualAge Generator Developer creates sample runtime JCL for VSE batch. The sample runtime JCL is based on JCL templates supplied with VisualAge Generator. The default templates shipped with VisualAge Generator are installed in the TEMPLATE subdirectory of the VisualAge Generator product directory.

The runtime JCL actually generated is based on the resources required by the program and the specific program file names. The examples referred to in this section were generated from the JCL templates shipped with VisualAge Generator.

Your system administrator can tailor the JCL that is generated by changing the JCL templates prior to generation. If you always need to make the same changes to the generated JCL, tailoring the templates can help reduce the number of manual changes required for the generated JCL.

Some of the reasons to tailor the JCL templates include:

- Implementing your installation's file-, naming conventions
- Setting up LIBDEF search chains
- Specifying a different DB2/VSE subsystem

Table 53 shows the relationship between the JCL templates used, the target environments, and the types of databases being used by the program. It also refers you to sample JCL.

Table 53. Runtime JCL Templates Based on Environment and Databases

JCL Template	Environment	Databases	Sample JCL
EFK0VEBE	VSE batch	No DL/I or DB2/VSE	Figure 61 on page 363
EFK0VEBD	VSE batch	With DB2/VSE, no DL/I	Figure 62 on page 365

Table 53. Runtime JCL Templates Based on Environment and Databases (continued)

JCL Template	Environment	Databases	Sample JCL
EFK0VEBC	VSE batch	With DL/I and DB2/VSE	Figure 64 on page 369
EFK0VEBB	VSE batch	With DL/I	Figure 63 on page 367
EFK0VEBA	VSE batch, called program	Any combination	DLBL statements Note: The contents of this member must be added to those program runtime job streams that call this program.

Table 54 shows the JCL templates that serve as models of DLBL statement generation for program-dependent files and databases.

Table 54. Model DLBL Statement for Program-Dependent Files and Databases

JCL Template	Contents
EFK0VSEI	VSE serial (file type SEQ) input file DLBL statement
EFK0VSEO	VSE serial (file type SEQ) output file DLBL statement
EFK0VVSI	VSAM input file DLBL statement
EFK0VVSO	VSAM output file DLBL statement
EFK0VDLI	DL/I Database DLBL statement for VSE batch
EFK0VCAL	Comment indicating where to insert DLBL statements for known transferred-to and called programs
EFK0VEZA	Comment indicating where to insert DLBL statements for programs transferred-to using the EZEAPP special function word
EFK0VEZD	Comment indicating where to insert DLBL statements for data sets dynamically allocated with the EZEDEST or EZEDESTP special function words

Refer to the *VisualAge Generator Generation Guide* for more information and for examples of tailoring JCL templates.

Modifying Runtime JCL

The sample runtime JCL for main batch programs contains EXEC statements to run the program. The JCL for main batch programs does not include JOB statements or DLBL statements for files used by called or transferred-to programs. Before you use the JCL to run the program, do the following:

- Add a JOB statement.
- Add the DLBL statements to the sample JCL for the main program for any files required by called or transferred-to programs (including those named with the EZEAPP special function word). Comments in the generated JCL indicate where to add the DLBL statements. You might also need to add DLBL statements for files used during run time with the EZEDEST or EZEDESTP special function words. You can also customize the sample runtime JCL with respect to specific file- name assignments, output file space allocations, additional steps, and other relevant data.

The type of runtime JCL generated for a main batch program varies based on the types of databases used by the main program, as shown in Table 53 on page 358 . The generated runtime JCL does not consider the types of databases used by called or transferred-to programs.

For example, consider the following situation:

- Program A is a main batch program that does not use relational databases.
- Application B is a main program that gains access to relational databases.
- Programs A and B are generated for the VSE batch environment.
- Program A transfers to program B using a DXFR statement.

Because program A does not use DB2/VSE, the JCL generated for program A is for a main batch program without DB2/VSE access (as shown in Figure 61 on page 363). This JCL will not work correctly because program B requires DB2/VSE to run. However, the JCL generated for program B is for a VSE batch job with DB2/VSE access (as shown in Figure 62 on page 365). The runtime JCL for program B can serve as a starting point for creating the JCL required to run program A. The following changes are required to the runtime JCL for program B:

- Change PREP=APPLB to PREP=APPLA on the // EXEC card for the DB2/VSE precompile.
- Add any DLBL statement for files required by program A or other programs in the job step.
- Ensure that the SID and SDB (optional) parameters are coded in the SYSIPT to the execution JCL for the main batch program. In this case, VisualAge Generator Server saves the information and defers the database connection to right before the first SQL call in the CALL or DXFR program. If no SID and SDB parameters are specified, the CALL EZECONCT service routine must be coded in the main program or the CALL or DXFR program before the first SQL I/O call.

If program B is a called program and program A calls B rather than using a DXFR statement, the runtime JCL for program B consists only of DLBL

statements. In this situation, you need to create your own program JCL. Any one of the following can serve as a starting point for the JCL:

- The runtime JCL for another main program that has access to relational databases
- The JCL template for the appropriate combination of DL/I and DB2/VSE
- The examples listed in Table 53 for the appropriate combination of DL/I and DB2/VSE

You can avoid the modification just described if you include a DB2/VSE process option in the initial main program.

If you get a JCL error for the runtime JCL, check the generation listing for the programs involved for any error messages related to JCL generation. In addition, ensure that the tailoring done for the JCL templates is correct. Also check any changes you made when you customized the sample runtime JCL.

Running Main Programs under VSE Batch

A main batch program generated for the VSE batch environment can be started by submitting the generated runtime JCL to the VSE/POWER reader queue. The runtime JCL is transferred to a VSE library by the VisualAge Generator program prepare function. The library member name is the program name. The member type is JCX.

If you modify the runtime JCL on the workstation, you can submit the modified JCX file to the VSE batch system by entering the following command in an OS/2 window:

```
SEND applname.JCX hostsession: ( FILE=RDR
```

where:

applname

Specifies the name of your program

hostsession

Specifies the letter representing the host session to which your batch program is attached.

The SEND command places the runtime JCL in the VSE/POWER RDR queue (FILE=RDR).

To receive job completion messages on the workstation, enter the following command in an OS/2 window:

```
RECEIVE applname.MSG jobname hostsession: ( FILE=PMSG
```

where:

applname

Specifies the name of the file on the workstation that contains the job completion messages.

jobname

Specifies the name of the job on the // JOB statement. This is optional. If *jobname* is not specified, the RECEIVE command retrieves all job completion messages that originate from your terminal.

hostsession

Specifies the letter representing the host session to which messages should be sent.

The RECEIVE command gathers messages from the VSE/POWER message service routine (FILE=PMSG). Then the RECEIVE command places the messages in a file with a name equal to the *applname* you specify and a file extension of MSG.

Refer to the *VSE/ESA Programming and Workstation Guide* for more information about these commands.

Called programs cannot be started with JCL. Called programs must be started by another VisualAge Generator program or by a non-VisualAge Generator program.

If you get a JCL error for the runtime JCL, check for any error messages related to JCL generation in the results of the generation for the programs involved. In addition, ensure the tailoring that was done for the JCL templates is correct. Also check any changes you made when you customized the sample runtime JCL.

Examples of Runtime JCL for VSE Batch Programs

The generated JCL in the following examples has these characteristics:

- The examples are based on the JCL templates shipped with VisualAge Generator. Your actual JCL templates might differ if your system administrator has tailored them for your organization. Refer to the *VisualAge Generator Generation Guide* for more information about tailoring JCL templates.
- Lowercase text appears in the examples where a generic example name has been substituted for an actual program or library name.

Running a Main Batch Program with No Database Access

Figure 61 shows the JCL used to start a main batch program. This example was generated with the runtime JCL templates provided with VisualAge Generator.

```
// JOB app1-name
// OPTION ERRS,JCANCEL,LOG
NOLOG
ON $ABEND GOTO $EOJ
ON $CANCEL GOTO $EOJ
ON $RC > 4 GOTO $EOJ
* Program specific DLBL statements
// DLBL filename-1,'file-id',.....
// DLBL filename-n,'file-id',.....
*
LIBDEF PROC,SEARCH=(PRD2.EZELIB),TEMP
SETPARM COBLIB='
SETPARM ELALIB='PRD2.&110e1a;B'
SETPARM LODLIB='your-vse-sublibrary'
SETPARM USRLIB='PRD2.EZELIB'
SETPARM PARM2='USRLIB,LODLIB,COBLIB,ELALIB'
// EXEC PROC=ELACUSTP,&PARM2
// LIBDEF *,SEARCH=(&LODLIB,&USRLIB,&ELALIB,&COBLIB),TEMP;
* -----*
// EXEC PGM=app1-name,SIZE=AUTO
/*
/ &
```

Figure 61. JCL for Main Batch Program Run as VSE Batch with no DL/I or DB2/VSE Access

Running a Main Batch Program with DB2/VSE Access

Programs that use DB2/VSE can run in multiple user mode or single user mode. Multiple user mode means that the DB2/VSE database can be shared with other jobs and CICS users running at the same time. Single user mode means that the job has a private DB2/VSE database.

You can modify the JCL to start a main batch program in single user mode by setting the user-defined symbol `SQLMODE` to `SINGLEUSER`. The default is `MULTIUSER`.

If you select to run your program in single user mode, you must modify the runtime JCL the following two conditions exists: if the main program does not include SQL statements, but calls or transfers to programs that do use SQL statements.

If the program does not explicitly connect to the database before the first SQL I/O process option, you must modify the generated runtime JCL to specify

the SQL user ID and password (*SID=userid/password*) and, optionally, the database server name (*SDB=database server name*) in the second line of the SYSIPT input file in the JCL. The JCL generated using the default template contains a SID and SDB specifications. The same SQL user ID, password, and database server name are used at preparation. If you plan to use a different SQL user ID and password for run time, you must change *userid* and *password* to your SQL user ID and password in the SID parameter. The SQL user ID and password are used to connect to the DB2/VSE database. You must specify your database server name in the SDB parameter if you do not want to connect to a default database server.

Figure 62 on page 365 shows the JCL used to start a main batch program that has access to DB2/VSE resources. This example was generated with the runtime JCL templates provided with VisualAge Generator.

```

// JOB appl-name
// OPTION ERRS,JCANCEL,LOG
NOLOG
ON $ABEND GOTO $EOJ
ON $CANCEL GOTO $EOJ
ON $RC > 4 GOTO $EOJ
* Program specific DLBL statements
// DLBL filename-1,'file-id',.....
// DLBL filename-n,'file-id',.....
*
LIBDEF PROC,SEARCH=(PRD2.EZELIB),TEMP
* -----*
*
*   To run in SQL Single User Mode, change SETPARAM SQLMODE= so
*   that it reads:
*
*           SETPARAM SQLMODE='SINGLEUSER'
*   Note: ELASQLDB.PROC must be customized to your installation.
* -----*
SETPARM SQLMODE='MULTIUSER'
* -----*
SETPARM COBLIB='
SETPARM ELALIB='PRD2.&110e1a;B'
SETPARM LODLIB='your-vse-sublibrary'
SETPARM USRLIB='PRD2.EZELIB'
SETPARM SQLLIB='
SETPARM PARM2='USRLIB,LODLIB,COBLIB,ELALIB,SQLLIB'
// EXEC PROC=ELACUSTP,&PARM2
* -----*
// LIBDEF *,SEARCH=(&LODLIB,&USRLIB,&ELALIB,&COBLIB,&SQLLIB),TEMP;
* -----*
IF SQLMODE NE 'SINGLEUSER' THEN
GOTO MULTUSR
// EXEC PROC=ELASQLDB          *--- Single User Database Definition
// EXEC PGM=ARISQLDS,SIZE=AUTO,PARM='SYSMODE=S,PROGNAME=appl-name'
SID=sqluserid/password,SDB=sql-database-name;
/*
GOTO JOBEND
/. MULTUSR
// EXEC PGM=appl-name,SIZE=AUTO
SID=sqluserid/password,SDB=sql-database-name;
/*
/. JOBEND
/&

```

Figure 62. JCL for Main Batch Program Run as VSE Batch with DB2/VSE Access

Running a Main Batch Program with DL/I Access

DL/I batch programs can run in single partition mode or multiple partition mode. Single partition mode means that the job has exclusive access to the DL/I databases. Multiple partition mode means that the job shares the database with other jobs or CICS transactions, and that the job gains access to the database through the DL/I MPS partition. The default JCL generated by

the templates provided with VisualAge Generator uses single partition mode. You might want to modify the templates or the generated runtime JCL to run in multiple partition mode. Refer to the DL/I DOS/VS application programming manual for more information on JCL for multiple partition support.

The JCL for the step that runs the batch program must include DLBL statements for the log if databases are opened with update intent, or if the program uses the VisualAge Generator CALL AUDIT service routine. Also, a DLBL statement must be included for each of the data sets associated with the DL/I databases referenced in the JCL.

The generation function uses the JCL template EFK0VDLI to build the DLBL statements for program databases. This template has the DLBL statement commented out because the high-level program database qualifiers are not collected during generation. You need to provide the final tailoring of these DLBL statements in the sample runtime JCL. Alternatively, depending on your naming conventions, your administrator might be able to modify the EFK0VDLI template so that you can use the /SYMPARM generation option to set high-level qualifiers for databases. Refer to the *VisualAge Generator Generation Guide* for information about modifying templates and using the /SYMPARM generation option.

Figure 63 on page 367 shows sample JCL used to run a generated program as a DL/I batch program. This example was generated with the runtime JCL templates provided with VisualAge Generator.


```

// JOB appl-name
// OPTION ERRS,JCANCEL,LOG
// UPSI 00000010
NOLOG
* *****
*      Set MPS to YES if running multiple partition support.      *
* *****
SETPARM MPS='NO'
* *****
ON $ABEND GOTO $EOJ
ON $CANCEL GOTO $EOJ
ON $RC > 4 GOTO $EOJ
* Program specific DLBL statements
// DLBL filename-1,'file-id',.....
// DLBL filename-n,'file-id',.....
*
// LIBDEF PROC,SEARCH=(PRD2.EZELIB)
SETPARM COBLIB='
SETPARM ELALIB='PRD2.&110e1a;B'
SETPARM DLILIB='
SETPARM LODLIB='your-vse-sublibrary'
SETPARM USRLIB='PRD2.EZELIB'
SETPARM PARM2='DLILIB,USRLIB,LODLIB,COBLIB,ELALIB'
// EXEC PROC=ELACUSTP,&PARM2
// LIBDEF *,SEARCH=(&USRLIB,&LODLIB,&ELALIB,&DLILIB,&COBLIB),TEMP;
// IF MPS EQ 'NO' THEN
// GOTO NOMPS
* -----*
// EXEC PGM=DLZMPI00,SIZE=256K      <--- For MPS Batch Support.
   DLI,appl-name,appl-psbname
/*
// GOTO EOJ
* -----*
/. NOMPS
// EXEC PGM=DLZRRC00,SIZE=256K      <--- For Batch Support, No MPS.
   DLI,appl-name,appl-psbname
/*
/. EOJ
/&

```

Figure 63. JCL for Main Batch Program Run as VSE Batch with DL/I Access

Running a Main Batch Program with DB2/VSE and DL/I Access

DB2/VSE databases must be accessed in multiple user mode. If the program does not connect to the database before the first SQL I/O process option, you must modify the generated runtime JCL to specify the SQL user ID and password (*SID=userid/password*) in the second line of the SYSIPT input file in the JCL. The JCL generated using the default template contains a SID and SDB specifications. The same SQL user ID, password, and database server

name are used at preparation. If you plan to use a different SQL user ID and password for run time, you must change *userid* and *password* to your SQL user ID and password in the SID parameter.

DL/I batch programs can run in single partition mode or multiple partition mode. Single partition mode means that the job has exclusive access to the DL/I databases. Multiple partition mode means that the job shares the database with other jobs or CICS transactions, and that the job gains access to the database through the DL/I MPS partition. The default JCL generated by the templates provided with VisualAge Generator uses single partition mode. You might want to modify the templates or the generated runtime JCL to run in multiple partition mode. Refer to the DL/I DOS/VS application programming manual for more information on JCL for multiple partition support.

Figure 64 shows the JCL that enables a program to run as a stand-alone DL/I batch processing program and to gain access to DB2/VSE databases. This example was generated with the runtime JCL templates provided with VisualAge Generator. Special recovery considerations are required. Refer to the DB2/VSE documentation for your system for additional information.

The generation function uses the JCL template EFK0VDLI to build the DLBL statements for DL/I program databases. This template has the DLBL statements commented out because the high-level program database qualifiers are not collected during generation. You need to provide the final tailoring of these DLBL statements in the sample runtime JCL. Alternatively, depending on your naming conventions, your administrator might be able to modify the EFK0VDLI template so that you can use the /SYMPARM generation option to set high-level qualifiers for databases. Refer to the *VisualAge Generator Generation Guide* for information about modifying templates and using the /SYMPARM generation option.

```

// JOB appl-name
// OPTION ERRS,JCANCEL,LOG
// UPSI 00000010
NOLOG
* *****
*      Set MPS to YES if running multiple partition.      *
* *****
SETPARM MPS='NO'
* *****
ON $ABEND GOTO $EOJ
ON $CANCEL GOTO $EOJ
ON $RC > 4 GOTO $EOJ
* Program specific DLBL statements
// DLBL filename-1,'file-id',.....
// DLBL filename-n,'file-id',.....
*
// LIBDEF PROC,SEARCH=(PRD2.EZELIB)
SETPARM COBLIB='
SETPARM ELALIB='PRD2.&110e1a;B'
SETPARM DLILIB='
SETPARM SQLLIB='
SETPARM LODLIB='your-vse-sublibrary'
SETPARM USRLIB='PRD2.EZELIB'
SETPARM PARM2='DLILIB,USRLIB,LODLIB,COBLIB,ELALIB'
// EXEC PROC=ELACUSTP,&PARM2
// LIBDEF *,SEARCH=(USRLIB,LODLIB,ELALIB,DLILIB,SQLLIB,COBLIB),TEMP
// IF MPS EQ 'NO' THEN
// GOTO NOMPS
* -----*
// EXEC PGM=DLZMPI00,SIZE=256K      <--- For MPS Batch Support.
  DLI,appl-name,appl-psbname
  SID=sqluserid/password SDB=sql-database-name;
/*
// GOTO EOJ
* -----*
/. NOMPS
// EXEC PGM=DLZRRC00,SIZE=256K      <--- For Batch Support, No MPS.
  DLI,appl-name,appl-psbname
/*
/. EOJ
/&

```

Figure 64. JCL for Main Batch Program Run as VSE Batch with DB2/VSE and DL/I Access

Recovery and Restart for Batch Programs

You should develop recovery procedures in the event of program or system problems. VisualAge Generator does not generate JCL for restart or recovery procedures.

Chapter 37. Moving Prepared Programs to Other Systems from VSE Systems

After you have prepared your program on one system you might want to move the program to a different production system.

Moving Prepared Programs to Another VSE System

If a program has been completely prepared on one system and you want to move the prepared program to another system, perform the following steps:

1. Copy the phases associated with the program, print services, table programs, and map group format module to a library on the other system.
2. Copy the runtime JCL for VSE batch programs to a library on the other system. Modify the JCL as necessary for the new system.
3. Use the DB2/VSE utilities to back up the DB2/VSE packages for SQL programs on the development system and reload them in the DB2/VSE database on the new system.
4. Follow the procedures identified in this manual for defining programs to VSE and CICS.
5. Define files and databases used by the program on the new system.

Maintaining Backup Copies of Production Libraries

Follow your installation-defined guidelines and procedures for making backup copies of production libraries. Having backup copies of production libraries enables you to return to the prior level of a program in case of errors.

You might also want to set up procedures for saving a copy of the program source in external or internal source format at the time a program is generated. Having a copy of the program source enables you to recreate production level libraries at any change level if necessary.

Part 8. Utilities for VSE Systems

Chapter 38. Using VisualAge Generator

Server Utilities on VSE Systems	375
Using the CICS Utilities Menu	375
New Copy	377
Diagnostic Message Printing Utility	380

Chapter 39. Diagnostic Control Options 383

Change or View Diagnostic Control Options for a Transaction	384
Change or View Default Diagnostic Control Options	387

Chapter 40. Using the Message File

Conversion Utility	389
Converting a Message File to an External Source Format Table.	389
Sample JCL for the Message File Conversion Utility	391

Chapter 41. Using the Parameter Group

Utility	393
--------------------------	-----

Chapter 38. Using VisualAge Generator Server Utilities on VSE Systems

VisualAge Generator Server provides the following utilities:

- A set of utilities in CICS to help manage the error diagnosis and control facilities of the VisualAge Generator Server runtime environment
- A batch utility for message file conversion
- A CICS utility to create and maintain parameter group files

You can access these utilities from the CICS utilities menu.

Using the CICS Utilities Menu

To access the CICS utilities do the following:

1. Log on to CICS.
2. Type ELAM on a clear screen.
3. Press Enter. When the ELAM transaction is started, a copyright panel is displayed.
4. Press Enter. The CICS Utilities Menu (Figure 65 on page 376) is displayed.

```
ELAM                               VisualAge Generator Server
                                   CICS Utilities Menu

Select one of the following utilities; then press Enter.

Action...._

_1. New Copy
_2. Diagnostic Message Printing
_3. Diagnostic Control Options

ENTER  F1=HELP  F3=EXIT
```

Figure 65. CICS Utilities Menu

Three functions are available from the CICS Utilities Menu panel (Figure 65):

New Copy

This function causes a new copy of a program, map group, or table to be used by subsequent transactions. Use the new copy function when programs, map groups, and tables are modified and generated again.

For programs and map groups, you can use the the VisualAge Generator Server new copy utility or the CICS NEWCOPY command to cause the new copy of the program to be used the next time a load request is issued for the program. The VisualAge Generator Server new copy utility does a new copy for both the online print services program and the map group format module when you specify a part type of Map Group. If you use the CICS NEWCOPY command for a map group, you must issue the NEWCOPY for both the online print services program and the map group format module.

For tables, you must use the VisualAge Generator Server new copy utility to cause a fresh copy of the table to be used the next time a load request is issued for the table. Do not use the CICS NEWCOPY command for tables.

Diagnostic Message Printing

This function routes the diagnostic messages in an error destination transient data queue to a spool file for printing or subsequent processing.

Diagnostic Control Options

This function lets you view or change the diagnostic control options set for the installation or for individual transactions. The options include dump control, error message routing to a transient data queue or the CICS journal, and transaction disabling when serious problems occur.

New Copy

The VisualAge Generator Server new copy utility causes a new copy of a program, map group, or table to be used by subsequent transactions. Transactions that are in progress when this function was started continue to use the copy that was current when the transaction began. Programs must end or reach the end of a segment before the new copy is used. The VisualAge Generator Server new copy utility must be run separately for programs, map groups, and tables to replace the copy already in storage.

To gain access to the VisualAge Generator Server new copy utility do the following:

1. Select option 1, New Copy, on the CICS Utilities Menu panel (Figure 65).
2. Press Enter.

The New Copy panel (Figure 66 on page 378) is displayed.

Note: You can also gain access to the VisualAge Generator Server new copy utility by doing the following:

1. Type ELAN on a clear screen.
2. Press Enter. When the ELAN transaction is started, a copyright panel is displayed.
3. Press Enter. The New Copy panel (Figure 66 on page 378) is displayed.

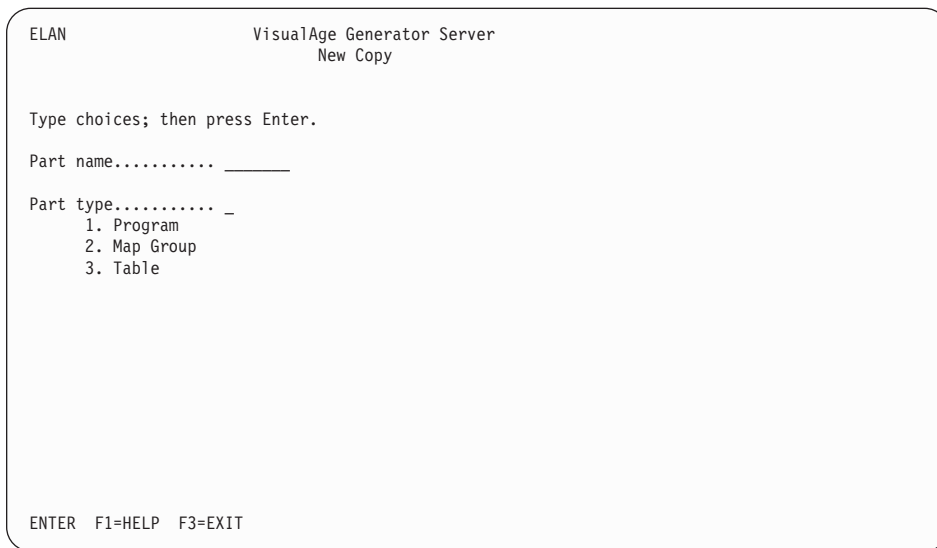


Figure 66. New Copy panel

Enter the following on the New Copy panel:

Part name

Specifies the name of the program, map group, or table to be used as a new copy in subsequent transactions

Part type

Specifies the type of part to be replaced

Note: VisualAge Generator Server does not validate the part type. You must specify the correct part type because different processing is required for programs, map groups, and tables. If you have problems in processing after using the VisualAge Generator Server new copy utility, try the VisualAge Generator Server new copy utility again to ensure you specified the part type correctly.

The type can be one of the following:

Program

This type causes the function to issue a CICS SET PROGRAM(name) NEWCOPY command to access a new copy of the program. This command does not cause a new copy for called programs that are statically linked with their caller.

Map Group

This type causes the function to issue a CICS SET PROGRAM(name) NEWCOPY command to get access to a new copy of the map group format module and the online print services program associated with the map group.

Table This type causes the utility to issue a CICS SET PROGRAM(name) NEWCOPY command to access a new copy of the table program, and sets a flag for VisualAge Generator Server indicating that the new copy of the table is to be used the next time a program loads the table contents.

If the table has been generated as a shared table, currently running transactions continue to use the old copy of the table while new transactions share the new copy of the table.

You can also access the new copy utility in batch mode. To invoke the batch new copy utility, link to program ELABNEW:

```
EXEC CICS LINK PROGRAM("ELABNEW")
COMMAREA(passed-parms)
LENGTH(174)
```

where the **passed-parms** record has the following structure:

Field	Length in Bytes	Type of Data	Description
NLS code	3	Character	NLS code identifying language
Part name	8	Character	name of program, map group, or table to be used as a new copy in subsequent transactions
Part type	1	Character	Type of part to be replaced: "1" Program "2" Map group "3" Table For more information, see the description for part type in the online help discussion of the new copy utility.
Return code	2	Binary	Return code from new copy
Message 1	80	Character	Message returned from new copy
Message 2	80	Character	Message returned from new copy

The following fields must be provided by the user:

- NLS code
- Part name
- Part type

The other fields are filled in by the new copy utility.

Any nonzero return code means that the new copy operation was not successful. If a nonzero value is returned in the **return code** field, check messages 1 and 2 for details indicating what error occurred.

Note: Message 2 is not always filled in. It may be blank.

Diagnostic Message Printing Utility

Diagnostic message printing allows you to route diagnostic messages in an error destination transient data queue to a VSE/POWER spool file for printing.

To gain access to the diagnostic message print utility do the following:

1. Select option 2, Diagnostic Message Printing, from the CICS Utilities Menu panel (Figure 65 on page 376).
2. Press Enter.

The Diagnostic Message Printing panel (Figure 67 on page 381) is displayed.

Note: You can also access the diagnostic message print function by doing the following:

1. Type ELAU on a clear screen.
2. Press Enter. When ELAU is started, a copyright panel is displayed.
3. Press Enter. The Diagnostic Message Printing panel (Figure 67 on page 381) is displayed.

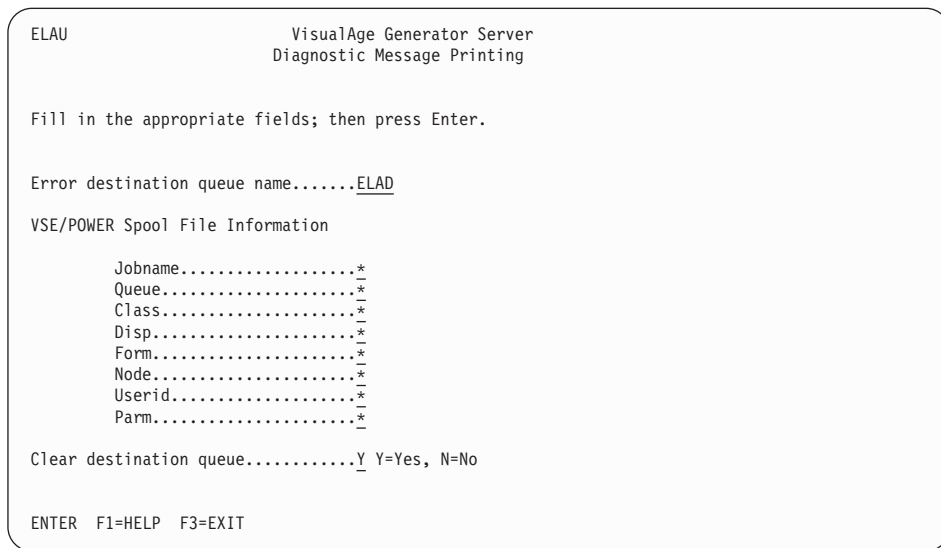


Figure 67. Diagnostic Message Printing panel

You can enter information in the following fields on the Diagnostic Message Printing panel:

Error destination transient data queue name

This field specifies the name of an existing error destination.

Enter the 1 to 4 character DCT name of the ERRDEST error destination transient data queue. The queue name is initialized to the default error destination queue. The default is ELAD.

VSE/POWER spool file information

The following fields specify the VSE/POWER LST queue member where the messages are to be written. Specifying an asterisk (*) in a field indicates that you want to use the system default for that value.

Jobname

1 to 8 characters that define the jobname for the VSE/POWER queue member. The default is EZEPRINT.

Queue

The destination VSE/POWER queue. Enter LST or PRT. The default is PRT.

Class A single character that specifies the print class. The default is A.

Disp The disposition of the VSE/POWER queue member once it is closed. Valid values are H (hold), K (keep), D (delete), or L (leave). The default is D.

- Form** 1 to 4 characters that identify the form number for print output. The default is your location's standard form.
- Node** 1 to 8 characters that specify the system node ID. The default is the current system node ID.
- Userid**
1 to 8 characters that specify the user ID. If you are signed on to VSE CICS (with either the CSSN transaction or the VSE/ESA interactive interface sign-on panel), the default is the value of the EZEUSRID special function word. If you are not signed on to VSE CICS, the default is an asterisk (*).
- Parm** A string of characters to be used to specify output operands for files on the VSE/POWER LST queue. This string is passed to CICS in the OUTDESCR option of the VSE CICS SPOOLOPEN OUTPUT command.

Clear destination queue

Specifies whether to clear the error queue of all messages after the messages are written to a spool file. The default is Y.

Refer to the *VisualAge Generator Design Guide* for more information on spool files on VSE CICS.

Chapter 39. Diagnostic Control Options

The diagnostic control options utility enables you to alter the diagnostic action options taken for a given transaction code that is assigned to a generated CICS program. If multiple transaction codes are assigned to a program, each transaction code is specified independently to the diagnostic control options utility.

You can also specify a default action to take place for transactions that are not explicitly defined to the diagnostic control options utility.

To gain access to the diagnostic control options utility, do the following:

1. Select option 3, Diagnostic Control Options, from the CICS Utilities Menu (Figure 65 on page 376).
2. Press Enter. The Diagnostic Control Options panel (Figure 68 on page 384) is displayed.

Note: You can also gain access to the diagnostic control options utility by doing the following:

1. Type ELAC on a clear screen.
2. Press Enter. When ELAC is started, a copyright panel is displayed.
3. Press Enter. The Diagnostic Control Options panel (Figure 68 on page 384) is displayed.

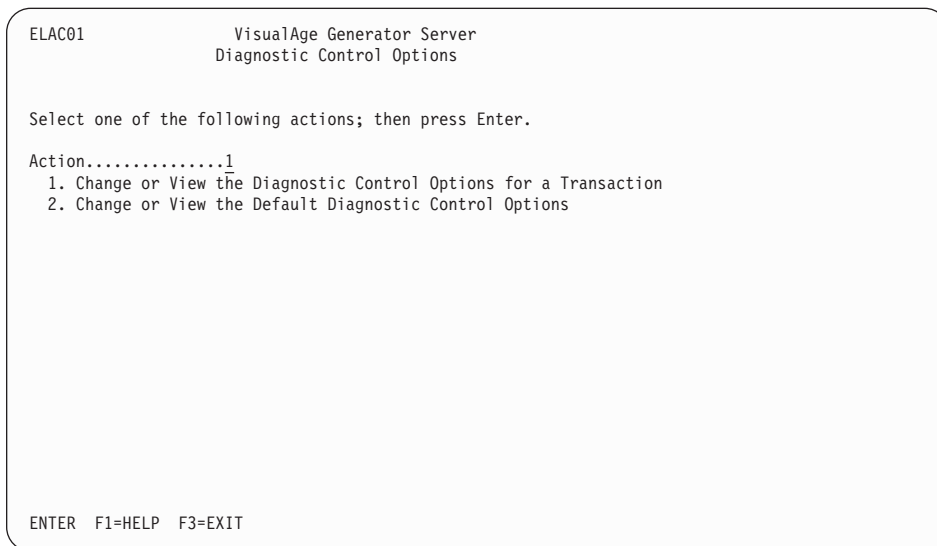


Figure 68. Diagnostic Control Options panel

You can gain access to the following functions from the Diagnostic Control Options panel:

Change or View the Diagnostic Control Options for a Transaction

This option enables you to change or view the diagnostic options for a specific transaction code.

Change or View the Default Diagnostic Control Options

This option enables you to change or view the installation default diagnostic options.

This affects transaction codes that are not specifically identified to the diagnostic controller.

Change or View Diagnostic Control Options for a Transaction

This function enables you to change the VisualAge Generator Server error diagnostic and control options in effect for a specific CICS transaction.

To start the function do the following:

1. Select option 1, Change or View the Diagnostic Control Options for a Transaction, from the Diagnostic Control Options panel (Figure 68).
2. Press Enter. The Change or View Diagnostic Control Options for a Transaction panel (Figure 69 on page 385) is displayed.

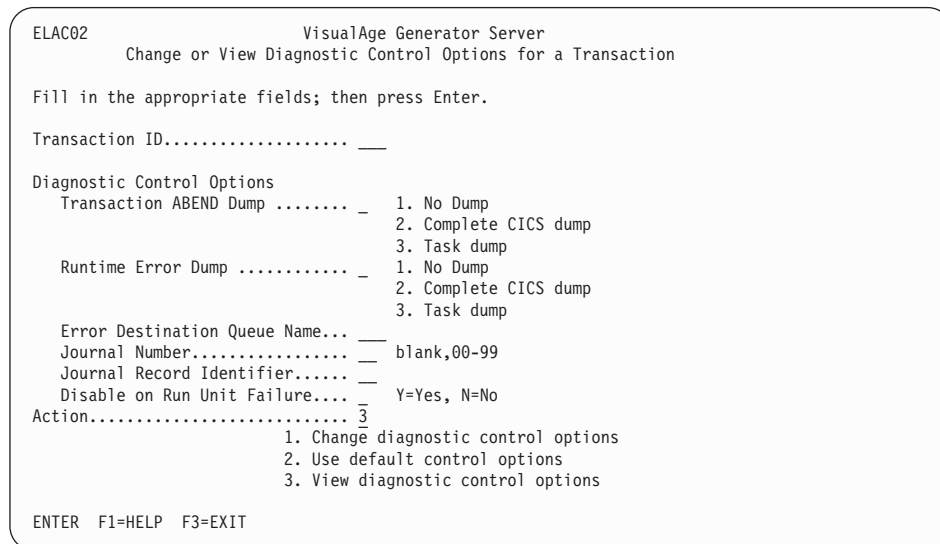


Figure 69. Change or View Diagnostic Control Options for a Transaction panel

The following fields can be entered on the Change or View Diagnostic Control Options for a Transaction panel :

Transaction ID

Specifies the 1 to 4 character identifier of the transaction you want to change the diagnostic options for

Diagnostic Control Options

Transaction ABEND Dump

Specifies the type of dump taken on a CICS transaction ABEND

The types of dumps are:

- 1. No Dump
- 2. Complete CICS dump
- 3. Task dump

Runtime Error Dump

Specifies the type of dump taken on a VisualAge Generator Server-detected error for which a dump is indicated in the error message explanation

The types of dumps are:

- 1. No Dump
- 2. Complete CICS dump

3. Task dump

Error Destination Queue Name

Specifies the 1 to 4 character name of a transient data queue to which VisualAge Generator Server error diagnostic messages are written whenever a transaction ends abnormally due to an error

If this field is blank, no messages are written to a queue.

Journal Number

Specifies the journal number of the CICS journal to which error diagnostic messages are written whenever a transaction is not successful due to an error

If this field is blank, no journal messages are written.

Journal Record Identifier

Specifies the 1 to 2 character record identifier used when messages are written to the CICS journal

If this field is blank, the default identifier EZ is used.

Disable on Run Unit Failure

Specifies whether a transaction is disabled whenever an error is detected that is likely to occur each time the transaction is run

Y Specifies that the transaction is disabled when these errors are detected

N Specifies that the transaction is not be disabled

Action

Allows you to change the current options, view the current options, or accept the default options

To change the options currently set for a transaction do the following:

1. Specify the transaction identifier and any changes.
2. Select 1, Change diagnostic control options.
3. Press Enter.

To use the installation defaults for the transaction do the following:

1. Type the transaction identifier.
2. Select 2, Use default control options.
3. Press Enter.

To view the options currently set for a transaction do the following:

1. Type the transaction identifier.

2. Select 3, View diagnostic control options.
3. Press Enter.

Change or View Default Diagnostic Control Options

This function enables you to change or view the default diagnostic options for transactions that are not identified to the diagnostic controller. If your default options were not modified at installation, the default diagnostic options are set as follows:

- Transaction ABEND and runtime errors both cause a task dump.
- The error destination queue name is ELAD.
- Diagnostic messages are not written to a CICS journal data set.
- Transactions are not disabled on a run unit error.

To start this function do the following:

1. Select 2, Change or View the Default Diagnostic Control Options, from the Diagnostic Control Options panel (Figure 68 on page 384).
2. Press Enter. The Change or View Default Diagnostic Control Options panel is displayed:

```

ELAC04                               VisualAge Generator Server
                                Change or View Default Diagnostic Control Options

Fill in the appropriate fields; then press Enter.

Default Diagnostic Control Options

Transaction ABEND Dump ..... 3   1. No Dump
                                      2. Complete CICS dump
                                      3. Task dump
Runtime Error Dump ..... 3       1. No Dump
                                      2. Complete CICS dump
                                      3. Task dump
Error Destination Queue Name... ELAD
Journal Number.....            b1ank,00-99
Journal Record Identifier..... EZ
Disable on Run Unit Failure.... N  Y=Yes, N=No

ENTER F1=HELP F3=EXIT

```

Figure 70. Change or View Default Diagnostic Control Options

The options on this panel are the same as those defined for changing or viewing the diagnostic control options for a transaction. They are all defined following Figure 69 on page 385.

Chapter 40. Using the Message File Conversion Utility

┌ General-Use Programming Interface _____

The message file conversion utility enables you to create an external source format message file from an existing message file. Once the message file has been converted to external source format, the file can be imported into VisualAge Generator to create the message table with its table contents. You can transfer message tables using the external source format import and export. Refer to the online help system provided with VisualAge Generator Developer for more information about the import and export facility.

└ End of General-Use Programming Interface _____

Converting a Message File to an External Source Format Table

┌ General-Use Programming Interface _____

Perform the following procedures to convert an existing message file into an external source format table:

1. Allocate file MSGFILE to an existing message file.
2. Allocate file MSGTBLE to a sequential file to receive the external source format. The file must have 80-byte fixed-length records.
3. Include the conversion parameter statement in your job stream. Figure 71 on page 390 shows the syntax of the conversion parameter statement.
4. Start the message utility, DZGMSGC.

└ End of General-Use Programming Interface _____

┌ General-Use Programming Interface _____

Once the external source format is created using the message file conversion utility, import the file. This will put the message table part into the library. You can then generate the tables to use when you run your programs. Figure 72 on page 392 shows sample JCL that can be used to run the conversion utility.

└ End of General-Use Programming Interface _____

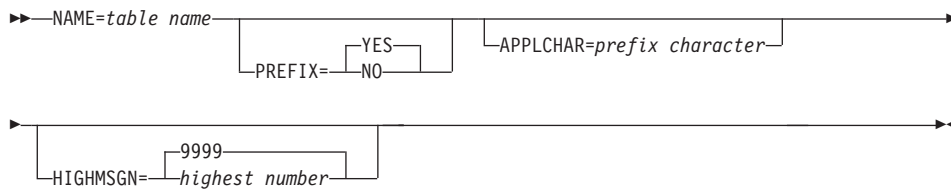


Figure 71. Message Table Conversion Parameters

NAME

Specifies the name of the table part to be created for VisualAge Generator Developer.

This keyword is required. The name can be 6 or 7 characters long, starting with an alphabetic character (A-Z). The remaining characters can be any combination of alphanumeric characters (A-Z, 0-9). The format of the name is vvvvccc, where vvvv is the value of the message table field specified in the program specifications, and ccc is a language code.

Valid language codes are as follows:

ENU	U.S. English
ENP	Uppercase English
CHS	Simplified Chinese
DEU	German
DES	Swiss German
ESP	Spanish
JPN	Japanese (Kanji)
KOR	Korean
PTB	Brazilian Portuguese

PREFIX

Specifies whether the prefix should be included in the message text.

- YES** Specifies that the prefix should be included in the message text. The default is YES.
- NO** Specifies that the prefix should not be included in the message text.

APPLCHAR

Specifies the character that is to be inserted before the message number. If a character is not specified, the prefix is created with a blank before the message number. In message files, this character

corresponds to the fourth character in the Message File field of the program specifications, or if there are less than 4 characters in this field, a blank.

HIGHMSGN

Specifies the highest message number to be converted. Valid values are 1 through 9999. The default is 9999. If the highest number is not known or you do not specify this parameter, the utility converts all messages in the message file.

The message conversion utility returns the following codes:

- 0** Message file conversion was successful
- 8** Error occurred during message conversion.

The return code is passed in register 15 through EZERCODE. This code can be checked for conditional execution of later steps in a job.

Sample JCL for the Message File Conversion Utility

In the sample JCL, the DITTO step near the middle of the JCL creates the MSGOPTS file containing the conversion parameter statement.

```

// JOB MSGCRUN 410 DELETE MISC
// ASSGN SYS007,SYSLST
* $$ LST LST=SYS007,DEST=(node,userid)
// ASSGN SYSIPT,SYSRDR
// ASSGN SYS004,SYSIPT
// OPTION PARTDUMP
// ID USER=vseid,PWD=xxxx
// UPSI 1
// ASSGN SYS005,ANYDISK,VOL=vvvvvv,SHR
// DLBL MSGOPTS,'user.MSGOPTS',0,SD
// EXTENT SYS005,vvvvvv,1,,cuu,len
* -----
* PLEASE NOTE: MSGOPTS file, You May Wish to Customize!!!
* -----
// EXEC DITTO
$$DITTO CSQ FILEOUT=MSGOPTS,CISIZE=2048
NAME=MSGENU PREFIX=YES APPLCHAR=A HIGHMSGN=9999
/*
$$DITTO EOJ
// DLBL MSGOPTS,'user.MSGOPTS',0,SD,BLKSIZE=80
// EXTENT SYS005,vvvvvv,1,,cuu,len
// DLBL MSGTBL,'user.MSGTBL.ESF',0,SD,BLKSIZE=88
// EXTENT SYS005,vvvvvv,1,,cuu,len
// DLBL usercat,'usercat',,VSAM
// DLBL MSGFILE,'user.MSGFILE.VSAMREL',,VSAM,CAT=usercat
* -----
* Run the Message Conversion Utility program (DZGMSGC)
* -----
// LIBDEF PHASE,SEARCH=(library)
// EXEC DZGMSGC,SIZE=AUTO,PARM='ENU'
/*
/&

```

Figure 72. Sample JCL for Message Conversion Utility

where:

- vseid** Specifies the VSE user ID
- xxxx** Specifies the VSE password
- vvvvvv** Specifies the volume ID
- user** Specifies the high-level qualifier for the data set
- cuu** Specifies the starting block, cylinder, or track location
- len** Specifies the number of blocks, cylinders, or tracks
- library** Specifies the VisualAge Generator library

Chapter 41. Using the Parameter Group Utility

Use the parameter group utility to create and maintain the parameter groups in the parameter group file. Each group contains parameters for controlling terminal printer utility (FZETPRT) transactions.

See “Special Parameter Group for the FZETPRT Program” on page 307 for a description of the startup parameters that can be included in the parameter group used with the FZETPRT program.

You can use the parameter group utility to perform the following operations:

- Display the contents of existing parameter groups
- View a list of existing parameter group names
- Add a new parameter group
- Change a parameter group
- Delete a parameter group

Table 55 shows the steps used to define a parameter group file.

Table 55. Defining Parameter Group Files

Procedure

VSE CICS

1. Define the parameter group file using the IDCAMS utility.
 DEFINE CLUSTER (NAME(PARM.GROUP.FILE)-
 RECORDS(100 100) KEYS(16 0) RECORDSIZE(272 272) INDEXED)
2. Initialize the parameter group file by using the IDCAMS REPRO function to insert a dummy record into the file.
3. Specify the FCT for the parameter group file utility to have access to a user-defined message file for VSE CICS.

```
DFHFCT TYPE=DATASET,           C
      DATASET=EZEPRMG,         C
      ACCMETH=VSAM,            C
      SERVREQ=(READ,UPDATE,ADD,DELETE,BROWSE), C
      FILESTAT=(ENABLED,CLOSED), C
      RECFORM=FIXED            C
```

4. Allocate the file by adding the following statement to the VSE CICS startup JCL:

```
// DLBL EZEPRMG, 'PARM.GROUP.FILE', ,VSAM
```

Table 55. Defining Parameter Group Files (continued)

Procedure

Note: The name that designates the parameter group file (EZEPRMG) is a reserved file name and cannot be used as a data file by a VisualAge Generator Developer-defined program.

When the file has been created and allocated, you can gain access the parameter group utility by doing the following:

1. Log on to CICS.
2. Type ELAP on a clear screen.
3. Press Enter.

The parameter group utility does not give message-specific tutorial help after a message is displayed and PF1 is pressed. See “Messages” on page 590 for more information and the user response for parameter group utility messages.

Once the parameter group utility has been started, the Parameter Group Specification panel (Figure 73) is displayed. You can specify the parameter group name on this panel.

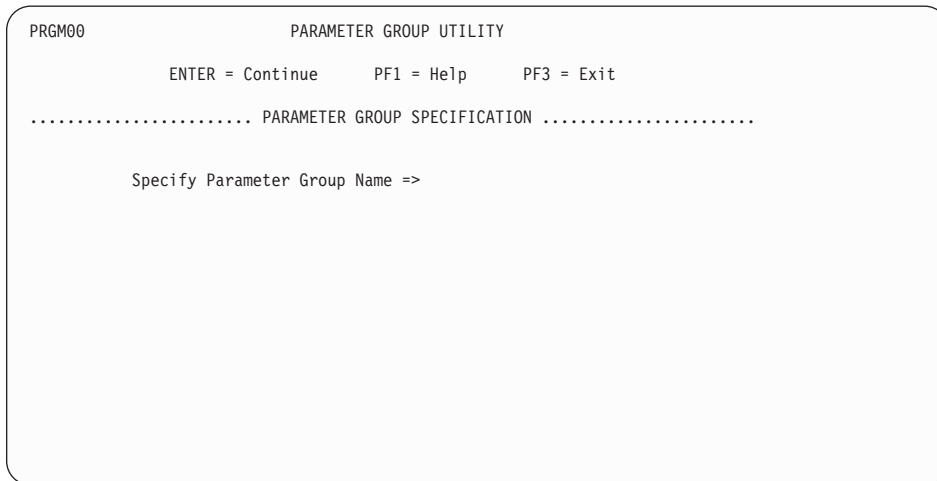


Figure 73. Parameter Group Specification panel

The parameter group name can be from 1 to 4 alphanumeric characters and must be the name of the transaction that was used to start the FZETPRT program. (The utility does not verify this.)

You can enter a group name that already exists if you want to modify a parameter group, or you can enter one that does not exist if you want to define a new parameter group.

Entering a question mark (?) as the group name on the Parameter Group Specification panel displays a list of previously-defined group names on the next panel, the Parameter Group List Display panel (Figure 74). Entering some characters followed by an asterisk (*) displays a list of parameter group names that begin with the characters that you entered. Entering a specific parameter group name displays the Parameter Group Definition panel (Figure 75 on page 396).

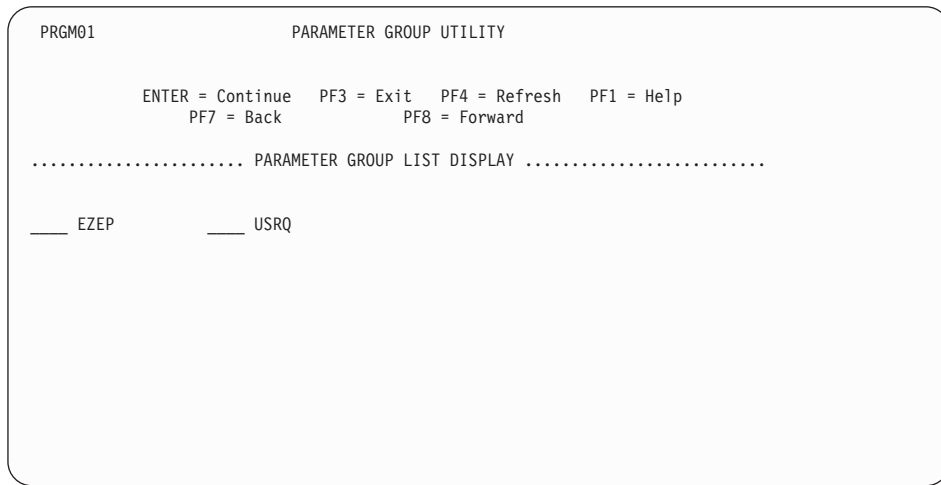


Figure 74. Parameter Group List Display panel

From the Parameter Group List Display panel, you can select a group name to edit by typing an S in the selection field to the left of the group name. You can delete a group by typing a D in the selection field to the left of the group name.

If the specified parameter group already exists, its contents are displayed on the Parameter Group Definition panel. The parameter group can be altered. If the specified parameter group does not exist, the Parameter Group Definition panel is displayed without any data. You can define the new contents; up to 256 characters of data can be entered for a parameter group.

```
PRGM02                PARAMETER GROUP UTILITY

                    PA2 = Cancel  PF1 = Help  PF3 = File and Exit
                    Parameter Group = CCCCCC

.....PARAMETER GROUP DEFINITION.....

Parameter Group:

=>M=PRTBUF=2048 PRTMPP=132 PRTTYP=D FORMFD=NO
```

Figure 75. Parameter Group Definition panel

The parameter group utility does not validate or format the parameters that are specified on the Parameter Group Definition panel. Any parameters that are not valid are ignored when the FZETPRT program is started. See “Special Parameter Group for the FZETPRT Program” on page 307 for more information about setting the parameters for terminal printing.

If you press PF3 on the Parameter Group Definition panel without entering any parameters, a parameter group is stored without any associated parameters. You can store an empty parameter group to reserve parameter group names.

Empty parameter groups do not affect the initialization of the FZETPRT program.

The parameter group utility left-justifies the parameter group name and pads it to the right with blanks (X'40'). The parameter group utility uses this name as a key to index the parameter group file.

If you selected a parameter group from the Parameter Group List Display panel (Figure 74 on page 395), after the Parameter Group Definition panel is processed, the Parameter Group List Display panel is displayed again with the original request replaced by an asterisk beside the group name that was processed. An asterisk (*) is ignored as input on the Parameter Group Definition panel if more processing is done.

Part 9. Diagnosing Problems on VSE Systems

Chapter 42. Diagnosing Problems for VisualAge Generator Server on VSE Systems

Systems	399
Detecting Errors	399
Reporting Errors	405
Controlling Error Reporting in VSE CICS Environments	405
Controlling Error Reporting in VSE Batch Environments	405
Error Reporting Summary	406
Using the VisualAge Generator Server Error Panel	408
Printing Diagnostic Information for VSE CICS	408
VSE CICS Diagnostic Message Layout	409
Running the Diagnostic Print Utility	410
Analyzing Errors Detected while Running a Program	410
Finding Information In Dumps	411
VisualAge Generator Server ABEND Dumps	411
COBOL or Subsystem ABEND Dumps	412
Information in the VisualAge Generator Server Control Block	413
Information in a Program, Print Services, or Table Profile Block	413
How to Find the Current Position in a Program at Time of Error	414
Using the VisualAge Generator Server Trace Facility	414
Enabling VisualAge Generator Program Source-Level Tracing with a Generation Option	415
Activating a Trace	415
Activating a Trace Session for VSE CICS	416
Activating a Trace Session for VSE Batch	419
Deactivating a Trace Session	422
Printing the Trace Output	422
VSE CICS	423
VSE Batch	423
Reporting Problems for VisualAge Generator Server	423
Problem Reporting Sequence	425

Chapter 43. Common Messages during Preparation for VSE Systems

VSE System and Job Control Messages	427
Workstation File Transfer Messages	427
DB2/VSE Precompiler Messages	428
COBOL Compilation Messages	429

Chapter 44. Common System Return Codes for VSE Systems

Common EZERT8 Return Codes	431
System Return Code Formats for EZERT8	432
Common System Return Codes in EZERT8	433
VisualAge Generator Server Return Codes	435
Common DB2/VSE Return Codes	445
Common DL/I Status Codes	447
Common VSAM Status Codes	448
OPEN request type	448
CLOSE request type	448
GET/PUT/POINT/ERASE/CHECK/ENDREQ request types	449
COBOL Status Key Values	449

Chapter 45. VisualAge Generator Server Return Codes and Abend Codes for VSE Systems

Return Codes	451
ABEND Codes	451
VSE CICS and VSE Batch Environments	451
VSE Batch Environment	454

Chapter 46. Codes from Other Products for VSE Systems

Common System Abend Codes for All Environments	457
LE Runtime Messages	458
Common COBOL Abend Codes	459
Common VSE CICS and VSE Batch Runtime Messages	459
Common VSE CICS and VSE Batch Abend Codes	461
COBOL Abends under CICS	462

Common Runtime Messages for DL/I
DOS/VS 462

Common Runtime Messages for SQL/DS 463

Chapter 42. Diagnosing Problems for VisualAge Generator Server on VSE Systems

This chapter contains diagnosis, modification, or tuning information. Use this information to determine the source of the problem you encountered. Some common program definition, database, and system errors that might cause problems are described. This chapter explains how to obtain error listings and diagnose runtime errors.

Detecting Errors

You can find most logic errors by using the test facility before you generate your program.

During generation, the VisualAge Generator Developer preprocessor checks your program for any remaining syntax errors. In addition, VisualAge Generator Developer also checks that your use of language elements is consistent with the resource association information you select for each file.

When you run your generated program, different types of errors are detected by VisualAge Generator Server, COBOL, the subsystem (VSE CICS or VSE batch), or the VSE operating system. The error handling varies depending on the product detecting the error. Table 56 shows the types of errors that can occur when you run your generated program, as well as VisualAge Generator Server functions you can use for error detection and reporting.

Table 56. Types of Errors

VisualAge Generator Set Controls	Error Detected By	Error Handling
----------------------------------	-------------------	----------------

Screen Variable Edit Errors:

The end user entered invalid data. For example:

- Alphabetic data in numeric field
 - Value not in allowable range
 - Omitted a required field
-

Use map definition to specify variable field edits or to include EZEMSG on the map.	Program logic or VisualAge Generator Server.	The error message is displayed on the map (in EZEMSG) and the cursor is placed on the field in error. If EZEMSG is not provided on the map, the ELAM01 screen is displayed with the EZEMSG error message.
---	--	--

File and Database Errors—Category 1:

A hard error occurred in a file or database I/O operation. For example:

- DL/I call not compatible with the processing options in the PSB
- I/O error on the file or database
- File not found

Use EZEFEFEC to indicate whether the program provides logic to override normal database and file error processing for hard errors. Hard errors are defined in the online helps within VisualAge Generator Developer.	Access method or database manager. The access method or database manager return an error code to the generated logic or VisualAge Generator Server.	If EZEFEFEC is set to 0 and a hard error occurs, the program ends with a transaction error, as described in Table 57 on page 406 . For SQL errors, the DB2/VSE code and error information are reported in the messages. Refer to the DB2/VSE messages and codes manual for an explanation of the error codes. If EZEFEFEC is set to 1 and a hard error occurs, the program is responsible for controlling the error handling.
--	---	--

File and Database Errors—Category 2:

A file or database error occurs due to incorrect program logic. For example, deleting or replacing a record without first doing an UPDATE process option is an example of this error type.

Follow the rules in the online documentation provided with the VisualAge Generator Developer product.	VisualAge Generator Server.	The program ends with a rununit error, as described in Table 57 on page 406.
---	-----------------------------	--

File and Database Errors—Category 3:

Two generated programs which are being used together are defined in an inconsistent manner. For example, two programs have been generated with different file definitions.

Follow the rules in the online documentation provided with the VisualAge Generator Developer product.	COBOL or VisualAge Generator Server.	If detected by VisualAge Generator Server, the program ends with a rununit error, as described in Table 57 on page 406. If detected by COBOL and the COBOL runtime option STAE is specified, a formatted dump is generated. If detected by COBOL and the COBOL runtime option NOSTAE is specified, a system dump is generated.
---	--------------------------------------	--

Arithmetic Overflow Errors:

One of three situations occurs:

- Maximum value overflow when the result of an arithmetic operation does not fit into the maximum size number (18 digits)
 - User variable overflow when the result of an arithmetic operation or a move to a numeric data item causes significant value (not decimal positions) to be lost
 - Division by 0
-

<p>Use the /NUMOVFL generation option to indicate that the generated program should check for overflow.</p>	<p>If /NUMOVFL is specified, special logic is generated to intercept numeric overflow conditions. The EZEOPER switch can be used to indicate the program provides logic to handle the overflow.</p>	<p>If /NUMOVFL is specified and the EZEOPER switch is set to 0 or 1, the maximum value overflow causes the program to end with a transaction error, as described in Table 57 on page 406 .</p>
<p>If /NUMOVFL is specified, use the EZEOPER special function word to indicate whether program processing should continue if an overflow occurs.</p>	<p>If /NONUMOVFL is specified, overflow errors are not detected. The contents of the EZEOPER special function word are ignored. Truncation occurs on maximum value overflow, or user variable overflow. An abend occurs on division by 0.</p>	<p>If /NUMOVFL is specified and the EZEOPER switch is set to 1, the user variable overflow also causes the program to end with a transaction error.</p>
		<p>In other situations, if /NUMOVFL, the program continues to run.</p>
		<p>If /NONUMOVFL is specified, overflow errors are not detected. Division by 0 causes an abend (0C7 for VSE batch and ASRA for VSE CICS systems). You need to code the program to ensure that overflow does not occur.</p>

Logic Errors:

The program ends because of faulty logic. For example, any of the following can cause a program to end:

- Numeric data that is not initialized
 - Problems with array subscripting
 - Infinite loops
 - Moving incompatible data
-

<p>Specifying the /INITRECD and /INITADDWS generation options forces numeric data in records to be initialized during program initialization. However, it is still possible that numeric data in files or databases contains nonnumeric values.</p>	<p>The test facility can be used to detect most of these errors before generation.</p> <p>If the COBOL STAE runtime option is specified, COBOL traps most of these errors.</p> <p>Otherwise, the VSE operating system detects the errors.</p>	<p>In VSE CICS, the EXEC CICS HANDLE ABEND function gives control to the generated program error handler. The error is then treated as a rununit error as described in Table 57 on page 406 .</p> <p>In VSE batch:</p> <ul style="list-style-type: none"> • If the COBOL runtime option STAE is specified, a formatted dump is generated. • If the COBOL runtime option NOSTAE is specified, a system dump is generated.
---	---	--

Program Interaction Errors:

Two generated programs which are being used together are defined or run in an inconsistent manner; or, all parts of a generated program were not installed correctly.

For example:

- A called program has been passed too few parameters.
- A program with a First Map specification is being transferred to using a DXFR statement.
- A transferred to program receives a map but does not have the First Map specification.
- The program linkage table did not specify the same linkage conventions when the calling and called programs were generated.

VSE CICS specific examples include the following:

- A program does a CONVERSE or DISPLAY process with a map, but the online print services program or the map group format module were not defined in the CICS PPT tables.
 - A program attempts to transfer to another program with a dynamic CALL or a DXFR statement, but the transferred to program was not defined in the CICS PPT tables.
 - A program attempts to gain access to a VisualAge Generator table, but the table program was not defined in the CICS PPT tables.
 - A program attempts to transfer to another program with an XFER statement, but the transferred-to transaction code was not defined in the CICS PCT tables.
-

<p>Follow the rules for the XFER and DXFR statements in the online documentation provided with the VisualAge Generator Developer product. For VSE CICS, follow the guidelines in Chapter 35. Preparing and Running Generated Programs in VSE CICS. For VSE batch, follow the guidelines in Chapter 36. Preparing and Running Generated Programs in VSE Batch. If special linkage conventions are used, ensure that the same linkage table is used for generating the calling and called programs.</p>	<p>VisualAge Generator Server. The program ends with a run unit error, as described in Table 57 on page 406.</p>
---	--

System Errors:

The program was not properly installed. Common mistakes are the following:

- Missing PSB or ACB
- JCL errors in batch programs
- Errors in CICS table definition
- Module cannot be found
- Storage cannot be obtained

<p>For VSE CICS, follow the guidelines in Chapter 35. Preparing and Running Generated Programs in VSE CICS. For VSE batch, follow the guidelines in Chapter 36. Preparing and Running Generated Programs in VSE Batch.</p>	<p>VSE CICS or VSE operating system.</p>	<p>The program ends with an CICS dump in VSE CICS systems or a system abend in VSE batch. For VSE CICS, dump production is controlled using the online diagnostic controller utility (“Controlling Error Reporting in VSE CICS” on page 405); for VSE batch systems, a system abend dump is produced if there are //OPTION DUMP or //OPTION PARTDUMP statements in the JCL.</p>
--	--	--

Reporting Errors

VisualAge Generator Server provides functions that help you determine the cause of a runtime problem. All runtime errors that VisualAge Generator Server traps are accompanied by error messages and supporting information to help diagnose the problem. Table 57 on page 406 shows the error diagnostic actions that can be taken based on the severity of the error and runtime system.

Controlling Error Reporting in VSE CICS

In the VSE CICS environment, error actions are controlled through the online diagnostic controller utility installed as transaction ELAC.

The utility allows you to specify what type of dump is requested, the name of the transient data queue to which VisualAge Generator Server diagnostic messages are written, the CICS journal number and identifier for error messages, and whether or not a transaction is disabled when a rununit error is detected. The utility lets you reset the default options for all transactions and override the default options for individual transactions.

See “Chapter 39. Diagnostic Control Options” on page 383 for more details about the diagnostic controller utility.

Controlling Error Reporting in VSE Batch Environments

The actions controlled by the runtime JCL are as follows:

- Print message. All output goes to the system logical unit SYSLST.
- SNAP dump. The output of the SNAP dump depends on the following conditions:
 - If the supervisor is running in 370 mode, the output always goes to the system logical unit SYSLST. However, if the system logical unit SYSLST is not assigned in the JCL, no dump output is produced.
 - If the supervisor is running in ESA mode, the output is routed to one of the following:
 - SYSLST if the // OPTION parameter in the JCL specifies NOSYSDMP
 - The dump data set for the partition if the // OPTION parameter specifies SYSDMP
- Abend 1600. This creates a dump and routes it to one of the following:
 - SYSLST if the // OPTION parameter in the JCL specifies NOSYSDMP
 - The dump data set for the partition if the // OPTION parameter specifies SYSDMP

This creates a dump only if the runtime JCL contains //OPTION DUMP or //OPTION PARTDUMP statements.

Error Reporting Summary

The following table summarizes the error processing actions for VisualAge Generator Server.

Table 57. Error Processing Actions For VisualAge Generator Server Detected Errors

Category of Error	Action
Transaction error - error affects only the current VSE CICS task. In VSE CICS, the transaction is still available to other end users.	VSE CICS: <ul style="list-style-type: none">• Write error messages to error destination (diagnostic controller option)• Write error messages to CICS journal data set (diagnostic controller option)• CICS dump, dump code ELAD, as determined by message. The type of dump issued for a particular transaction is a diagnostic control option.• Issue a rollback request• Display error messages on terminal, if possible• Set return code to 693 (called programs only) VSE Batch: <p>See run unit error.</p>

Table 57. Error Processing Actions For VisualAge Generator Server Detected Errors (continued)

Category of Error	Action
Rununit error - The error is likely to occur for every user. In VSE CICS, the transaction might be disabled.	<p>VSE CICS:</p> <ul style="list-style-type: none"> • Write error messages to error destination (diagnostic control option), if possible • Write error messages to VSE CICS journal data set (diagnostic control option), if possible • Disable transaction (diagnostic control option) • CICS dump, dump code ELAD, as determined by message. The type of dump issued for a particular transaction is a diagnostic control option. • Issue a rollback request • Display error messages on terminal, if possible • Set return code to 693 (called programs only) • Return <p>VSE BATCH:</p> <ul style="list-style-type: none"> • Print message (SYSLST) • SNAP dump as determined by the message and the // OPTION card in the runtime JCL • Issue a rollback if SQL databases used • Set return code to 693 • Return
Catastrophic error - storage is corrupted or standard error reporting processing is not successful.	<p>VSE CICS:</p> <ul style="list-style-type: none"> • Write error messages to error destination (diagnostic control option), if possible • Write error messages to CICS journal data set (diagnostic control option), if possible • Disable transaction (diagnostic control option) • Display error messages on terminal, if possible • ABEND ELAE. The type of dump issued for a particular transaction is a diagnostic control option. <p>VSE BATCH:</p> <ul style="list-style-type: none"> • Print message (SYSLST), if possible • Abend 1600 (SYSLST, if // OPTION DUMP or // OPTION PARTDUMP specified)
VisualAge Generator Server error at a point where standard error reporting process is not active	<p>All environments</p> <ul style="list-style-type: none"> • Ends abnormally, ABEND code indicates the reason.

See Table 59 on page 411 for information concerning the contents of the registers when either a 1600 or an ELAE abend occurs.

Using the VisualAge Generator Server Error Panel

When an error occurs, VisualAge Generator Server attempts to display error messages on the current terminal. The panels used in displaying error messages are defined as map group ELAxxx where xxx is the language code. External source format for the map group can be found in PRD2.ELA.V1R2M0, so that you can customize the constant information in the map definition to display error reporting instructions specific to your own installation. For example, you can add the phone number of your system administrator so that the user encountering the error knows who to call to report the problem.

The following figure shows the error panel (map ELAM02 in the map group) as it is shipped with the product. The panel displays the same error messages that are written to the error destination queue, system log or journal, or the system logical unit SYSLST. If there are more error messages than can fit on a single panel, the last line on the panel prompts the user to press a key to display additional error messages.

```
Unexpected Program Failure

An unexpected input/output or program error occurred in the
program you were running. Please make a note of the program
name, date, time, and initial error messages and report them to your
system administrator.

Program name ... ART22
Date ..... 08/21/90
Time ..... 13:04:23

Error Messages:

ELA00093I Error occurred in program ART22, process or group ART229
ELA00131P MSGQ error, file = UNKNOWN, function = CHG, status code = A1
ELA00066I DL/I I/O area = UNKNOWN
                        EDDDED4
                        45256650
```

Printing Diagnostic Information for VSE CICS

Diagnostic messages are sent to a transient data queue for CICS transactions. A diagnostic print utility is provided to print messages written to a transient data queue. Optionally, as specified by the diagnostic controller utility, the diagnostic information can also be written to an CICS journal data set.

VSE CICS Diagnostic Message Layout

Table 58 shows the format of the information in each error message record written to a transient data queue or CICS journal.

Table 58. Diagnostic Message Layout

Field	Length in Bytes	Type of Data	Description
SYSID name	4	Character	The name of the VSE CICS system that the error message was created on.
TRANID name	4	Character	The name of the CICS transaction code that started the logical unit-of-work.
Task identifier	8	Character	The task identifier assigned by CICS to each transaction instance that is processed. This number is reset to 0 when CICS is cold-started. This is taken from EIB field EIBTASKN.
ERRDEST name	4	Character	The name of the CICS transient data queue. This field is blank if the record is written to the CICS journal.
Date	8	Character	Date of the transaction (MM/DD/YY)
Time	8	Character	Time of the transaction (HH:MM:SS)
NLS	3	Character	The value specified for the /TARGNLS generation option
Message number	9	Character	The message number: Bytes 1-3 Message File Identifier (ELA) Byte 4 Program Identifier (0) Bytes 5-8 Message Number Byte 9 Message Type. A message type of 'C' indicates this record is a continuation of the specified message from a previous record in the queue.

Table 58. Diagnostic Message Layout (continued)

Field	Length in Bytes	Type of Data	Description
Message number separator (reserved position)			Byte 10 Blank
Message text	110	Character	The text from the message file with specified message inserts

Running the Diagnostic Print Utility

Use the ELAU transaction to print the messages routed to a transient data queue. See “Diagnostic Message Printing Utility” on page 380 for more information on running the diagnostic message printing utility and for a full description of the required data and their system defaults.

Analyzing Errors Detected while Running a Program

Use the error messages, diagnostic messages, and the information in “Chapter 45. VisualAge Generator Server Return Codes and Abend Codes for VSE Systems” on page 451 to determine the cause of the problem. If the error is detected by another product (for example, COBOL), check the information in Chapter 44. Common System Return Codes for VSE Systems and Chapter 46. Codes from Other Products for VSE Systems and the documentation for the other product. If you cannot determine the cause of the problem using this information and if the problem can be created again in the test environment, use the test facility to isolate and debug the errors before generating your program. For example, if your program attempts to replace a SQL record without first issuing a UPDATE or SETUPD call, the test facility can help you follow the flow of your program to determine where the problem is occurring.

For environmental debugging, you can use the runtime diagnostic facility (EDF) for CICS programs. In addition, if you use the TEST COBOL compile option, you can use the COBOL debugging facilities.

For more information about testing with the VisualAge Generator test facility, refer to the online documentation provided with the VisualAge Generator Developer product. Refer to the VSE CICS, and COBOL manuals for your versions of these products for additional information on their debugging facilities.

If you get a JCL error for the runtime JCL, check the generation listing for the programs involved for any error messages related to JCL generation. In

addition, ensure the tailoring that was done on the JCL templates and VisualAge Generator Server procedures is correct. Also check the changes you made when you customized the sample JCL.

When abends occur, the problem determination might require assistance from the IBM Support Center. In this case, be prepared to provide IBM with the following information:

- VisualAge Generator Developer print of the problem program
- COBOL compilation listing created using the /COMMENTLEVEL=STATEMENTS generation option
- Formatted dump
- VisualAge Generator Server diagnostic information written to either the error diagnostic queue or listed in the printout on the system logical unit SYSLSST
- VSE CICS journal, as appropriate

IBM requests a COBOL debugger trace listing only if the information is needed for problem determination. IBM gives you the information on how to specify the trace options if the information is necessary.

Finding Information In Dumps

Information about the problem program can be determined by finding the address of the VisualAge Generator Server control block in a dump.

VisualAge Generator Server ABEND Dumps

If the dump code is 1600, 1602, or ELAE, the dump was initiated because VisualAge Generator Server detected an error. Register 2 at ABEND points to the VisualAge Generator Server control block. Register 4 points to a linked list of messages formatted as shown in Figure 76 on page 412.

Table 59. Registers when a SNAP dump is taken or a VisualAge Generator Server ABEND occurs.

Reg.	Value
2	Points to VisualAge Generator Server control block. At offset 272 (hexadecimal offset 110) from the start of the VisualAge Generator Server control block is the address of the initial program profile block, which provides information about the first generated program that was started. At offset 276 (hexadecimal offset 114) from the start of the VisualAge Generator Server control block is the address of the current program profile block, which provides information about the VisualAge Generator program that was running at the time of the abend.
4	Points to the message buffer that contains all messages.

The following diagram shows the format of the message buffer that contains all the messages in the dump.

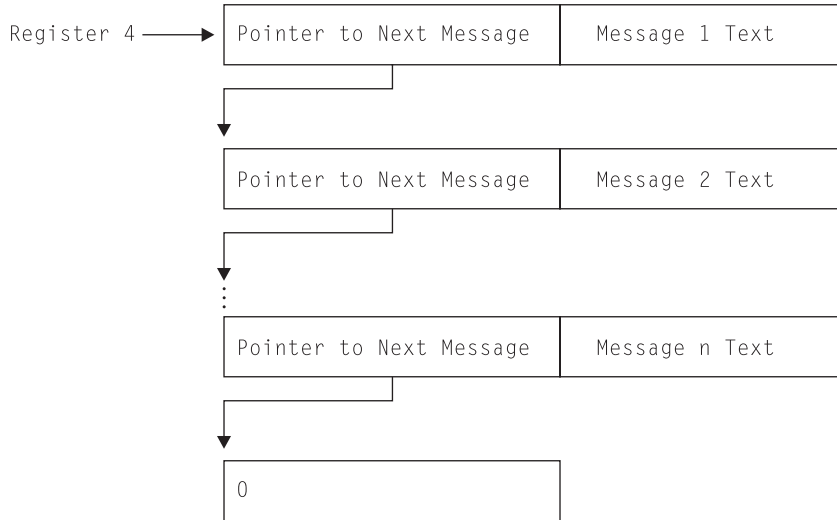


Figure 76. Message Buffer Format

COBOL or Subsystem ABEND Dumps

If the dump is not a VisualAge Generator Server abend, you can use the following method to locate the VisualAge Generator Server control block:

- On VSE CICS systems, locate the CICS Task Work Area (TWA) in the dump. Locate the string *EZERTS-CONTROL* in the TWA. This string is the identifier at the start of the VisualAge Generator Server control block. The * and - characters might be converted to periods in a formatted dump.
- On other systems, locate the string ELARHAPP followed immediately by a program name. ELARHAPP is the identifier at the start of a program profile block. The four-byte address at hex offset 20 in the program profile is the VisualAge Generator Server control block address. If 0, the program might not yet be activated. Do a search for another ELARHAPP control block followed by a program name. See Table 61 on page 413 for information in the program profile block.

Information in the VisualAge Generator Server Control Block

The following information is in the VisualAge Generator Server control block:

Table 60. Information in the VisualAge Generator Server Control Block

Offset in hex	Length in bytes	Contents
0	16	Control block identifier - *EZERTS-CONTROL*
104	4	CICS EXEC Interface Block (EIB) Pointer
110	4	Program profile address for current program
114	4	Program profile address for initial program
118	8	Terminal identifier
120	8	User identifier
128	8	Transaction identifier
150	12	EZEDLPSB
1CC	18	Current process or group

Information in a Program, Print Services, or Table Profile Block

Each generated COBOL program contains a profile control block in COBOL working storage initialized with information about the program. The first eight bytes contain an eye-catcher constant identifying whether the program is a program, print services program, or table program. The second eight bytes contain the program name. Other information in the profile block is shown in the following table:

Table 61. Locator Format for Generated COBOL Program Dumps

Offset in hex	Length in hex	Contents
00	08	Program type identifier: ELARHAPP — Application program ELAAHMGC — Print services program ELARMTTP — Table program
08	08	Program name
10	08	Program generation date (MM/DD/YY)
18	08	Program generation time (HH:MM:SS)
20	04	VisualAge Generator Server control block address
24	02	VisualAge Generator Developer version

Table 61. Locator Format for Generated COBOL Program Dumps (continued)

Offset in hex	Length in hex	Contents
26	02	VisualAge Generator Developer release
28	02	VisualAge Generator Developer modification level
2A	10	Reserved
34	08	Target runtime system

How to Find the Current Position in a Program at Time of Error

The VisualAge Generator Server control block identifies the currently running program and process or group at the time of the problem (Table 60 on page 413). Associated error messages identify the VisualAge Generator statement number for errors detected by VisualAge Generator Server that need statement identification to resolve the problem. For performance reasons, the generated COBOL program does not keep track of the VisualAge Generator statement number for each generated statement. If a program exception occurs in a generated program, you can determine the VisualAge Generator statement number by finding the COBOL statement that did not process correctly in a COBOL program listing that contains the VisualAge Generator statements generated as comments.

Using the VisualAge Generator Server Trace Facility

The VisualAge Generator Server trace facility can be used by the IBM Support Center to aid in problem determination, or by the program user to trace program activity.

There are two levels of tracing available:

- VisualAge Generator program source-level tracing
- VisualAge Generator Server runtime-level tracing

With source-level tracing, you can request traces of VisualAge Generator statements, traces of the data and error codes after every SQL call in a program, except SQL calls made with the SQLEXEC process option. Source-level tracing is enabled with the use of the /TRACE generation option. Source-level tracing is activated in the runtime environment by specifying trace filter criteria. See “Activating a Trace” on page 415 for more information on activating traces.

With runtime-level tracing, you can request a data stream trace, a VisualAge Generator Server internal dump trace, or a service routine trace. Runtime-level tracing does not require the use of a generation option. Runtime-level tracing

is activated in runtime environment by specifying trace filter criteria. See “Activating a Trace” for more information on activating traces.

Use these functions only with the assistance of the IBM Support Center. If you use these functions without assistance, large amounts of trace output might be produced based on trace option selection.

Enabling VisualAge Generator Program Source-Level Tracing with a Generation Option

You must specify the /TRACE generation option in order to get source-level trace output. The VisualAge Generator Developer preprocessor validates the generation option and its parameters. VisualAge Generator Developer creates the necessary COBOL code to accomplish the type of tracing that you request. The /TRACE generation option can be specified with these parameters:

NONE

VisualAge Generator program source-level tracing is not enabled

NONE is the default option. /TRACE=NONE excludes any other /TRACE generation options.

SQLERR

Enables tracing of only the SQL error codes returned in the SQL Communications Area (SQLCA) after an SQL call

SQLIO

Enables tracing of both the data and the SQL error codes returned in the SQL Communications Area (SQLCA) after an SQL call

Once a program has been generated with the /TRACE=SQLIO generation option, you can then turn on the SQLIO trace option, the SQLERR trace option, or both, or neither.

STMT Enables tracing of each VisualAge Generator statement that is encountered during the running of the program

Refer to the *VisualAge Generator Generation Guide* for more information on generation options.

Activating a Trace

Tracing is activated during run time in the VSE CICS environment by using the ELAZ transaction, and in the VSE batch environment by specifying trace selection information in the job stream following the EXEC statement (SYSIPT file).

Activating a Trace Session for VSE CICS

A utility is supplied to activate tracing in the VSE CICS environment. To start the utility, enter the utility transaction code, ELAZ. The utility transaction must start prior to running the transaction to be traced.

The ELAZ transaction must run in the same partition as the transactions to be traced. In VSE CICS, enter the ELAZ transaction and the transaction to be traced from terminals attached to the same VSE CICS partition.

Figure 77 shows the initial panel for the ELAZ transaction that enables you to specify which transactions are to be traced. You use a secondary panel to specify filter criteria for a specific transaction that control what information is traced for that transaction.

```
ELAZ01                               VisualAge Generator Server
                                     Trace Transaction Selection

Specify the transaction you want to trace; then press Enter.

To select specific programs and services for tracing, place the cursor
on a transaction name and press F4.

Transaction codes or initial program names

_____  _____  _____  _____
_____  _____  _____  _____
_____  _____  _____  _____
_____  _____  _____  _____

ENTER F1=HELP F3=EXIT F4=FILTER F9=REFRESH F10=STOP TRACE
```

Figure 77. VisualAge Generator Server Trace Transaction Selection Panel

VisualAge Generator Server then presents the panel shown in Figure 78 on page 417 for trace filter selection.

```

ELAZ02                VisualAge Generator Server
                       Trace Filter Selection

Transaction Code or Initial Program _____

Fill in the appropriate fields, then press Enter.

3270 Data Stream.....N      APP Statement Trace.....N
Terminal ID.....           SQL/IO Trace.....N
Trace to File.....N         SQL/ERR Trace.....N
VisualGen IDUMP Trace..N

FILENAME ELATnnnn  NODE   *   USERID  EZEUSRID  CLASS  A  FORM  *
-----
Programs
-----
Services
-----
-----
-----

ENTER  F1=HELP  F3=RETURN  F9=REFRESH

```

Figure 78. VisualAge Generator Server Trace Filter Selection Panel

The filter criteria include the following:

3270 Data Stream (Y or N)

Specifies whether to trace 3270 data streams

If yes (Y), the 3270 data streams built or received by VisualAge Generator Server are traced. The default is no (N).

Terminal ID

Specifies a terminal identifier

If specified, only transactions initiated from that terminal are traced. If not specified, service requests from any terminal are traced.

Trace to File (Y or N)

Specifies whether the trace output goes to a file

If yes (Y), all trace output is sent to a VSE/POWER LST queue member in addition to being sent to the in-storage wrap-around trace buffer. The VSE/POWER LST queue member has the name ELATnnnn where nnnn is the VSE CICS transaction ID.

Note: Two users cannot concurrently trace the same transaction because the ELATnnnn file is created as a resumable report. Use the terminal ID as a filter to prevent problems.

You must specify Trace to File as Y (yes) if the SQLIO Trace or SQLERR Trace are specified as Y (yes). Trace output for SQL I/O and SQL errors is always directed only to a VSE/POWER LST queue.

If no (N), the trace output (except for SQL I/O and SQL error tracing) is sent to an in-storage wrap-around trace buffer. The default is N. The size of this trace buffer is defined during customization of VisualAge Generator Server.

VisualGen IDUMP Trace (Y or N)

Specifies whether to dump VisualAge Generator Server internal storage areas

If yes (Y), the trace facility provides dumps of certain VisualAge Generator Server internal storage areas. The default is no (N), no internal storage dumps.

APP Statement Trace (Y or N)

Specifies whether to trace VisualAge Generator statements in a program

If yes (Y), the trace facility provides the process name and the statement for each VisualAge Generator statement that the program processes. You must use the /TRACE=STMT generation option to enable this type of tracing. The default is no (N).

SQL/IO Trace (Y or N)

Specifies whether to trace SQL I/O

If yes (Y), the trace facility provides traces of the data and error codes returned after an SQL call. The VisualAge Generator SQL process name, the process option, the process object, the SQL function name, and the VisualAge Generator data item name, length, type, and contents are given. You must use the /TRACE=SQLIO generation option to enable this type of tracing. The default is no (N).

SQL/ERR Trace (Y or N)

Specifies whether to trace SQL error information

If yes (Y), the trace facility provides traces of the error information that comes back from SQL/DS on every database call. The SQLCODE, SQLERRP, SQLSTATE, SQLWARN, SQLERRD, SQLEXT, and SQLERRMC codes are given. You must use the /TRACE=SQLERR or the /TRACE=SQLIO generation option to enable this type of tracing. The default is no (N).

FILENAME

VSE/POWER queue name for the trace output is ELATnnnn, where nnnn is the CICS transaction or the first four characters of the program name that originated the trace entry.

NODE

1 to 8 characters that specify the system node ID. The default is the current system node ID.

USERID

1 to 8 characters that specify the user ID. If you are signed on to VSE/CICS (with either the CSSN transaction or the VSE/ESA interactive interface sign-on panel), the default is the value of the EZEUSRID special function word. If you are not signed on to VSE CICS, the default is an asterisk (*).

CLASS

A single character that specifies the print class. The default is A.

FORM

1 to 4 characters that specify the form number for print output. The default is your location's standard form.

Applications

Specifies whether to limit the trace to certain application programs or print services programs

If specified, only the requested programs are traced.

Services

Specifies whether to limit the trace to certain services

If specified, only the requested services are traced. Otherwise all service numbers are traced if the other criteria are met.

Note: The entry to ELARSINI (initialization service) and the exit from ELARSTRM (cleanup service) are not traced. ELARSINI initializes the trace facility. ELARSTRM ends the trace facility.

If you are running a trace to aid in problem determination, enter the filter criteria as directed by the IBM support center.

Activating a Trace Session for VSE Batch

In VSE batch, tracing is activated by having the trace selection information in the job stream following the EXEC statement (SYSIPT file). The parameters must be followed by /*.

Both trace selection information and DB2/VSE SID and SDB parameters are from SYSIPT. The SID and SDB parameters must come first in the job stream, followed by the trace control statements. Two semi-colons (;;) on a separate line must precede any data records provided in the SYSIPT area. See "Using Sequential Data Files in VSE Batch" on page 325 for more information about the requirements for input records provided in the system logical unit SYSIPT. Figure 79 on page 420 shows the correct syntax for the trace control statements.

```

// EXEC progname
:FILTER DATASTREAM=Y|N
:FILTER TRACETOFILE=Y|N
:FILTER APPSTMT=Y|N
:FILTER SQLIO=Y|N
:FILTER SQLERR=Y|N
:FILTER IDUMP=Y|N
:APPLS

:
[ name ]

:
: EAPPLS
: SERVICES

:
[ service number ]

:
: ESERVICES
: EFILTER
/*

```

Notes:

1. Only one program name or service number can be entered on each line.
2. The :FILTER and :EFILTER tags are required if any other tags are included.
3. More than one filter can be specified on a line. The filters must be separated by 0 or more blanks. The example below shows sample :FILTER statements that are valid and equivalent:

```

:FILTER APPSTMT=Y
:FILTER SQLERR=Y

:FILTER APPSTMT=YSQLERR=Y

:FILTER APPSTMT=Y SQLERR=Y

:FILTER APPSTMT=Y          SQLERR=Y

```

The filters cannot be continued on the next line. The statement shown below is not valid:

```

:FILTER APPSTMT=Y SQLERR=
Y

```

Figure 79. Instream Filter Selection for VSE Batch

The control card tags and attributes that control filtering have the following meaning:

:FILTER

Options controlling what information is traced and where trace data is written

The following attributes can be used with the `:FILTER` statement:

- `DATASTREAM=Y|N`

If `DATASTREAM=Y` is specified, the 3270 data streams built or received by VisualAge Generator Server are traced. The default value is `N`, no data stream tracing.

- `TRACETOFILE=Y|N`

If `TRACETOFILE=Y` is specified, the trace output is directed to a VSE/POWER LST queue named `ELATOUT`.

If `TRACETOFILE=N` is specified, the trace output goes to an in-storage wrap-around trace buffer. The size of this trace buffer is defined during customization of VisualAge Generator Server. The default for the `TRACETOFILE` option is `N`.

If `TRACETOFILE=Y` is specified, all trace output is sent to the VSE/POWER LST queue in addition to being sent to the in-storage wrap-around buffer.

You must specify `TRACETOFILE=Y` if the `SQLIO=Y` or `SQLERR=Y` are specified. All trace output for SQL I/O and SQL errors is directed only to a VSE/POWER LST queue, not to the in-storage wrap-around trace buffer.

- `APPSTMT=Y|N`

If `APPSTMT=Y` is specified, the trace facility provides the process name and the statement for each VisualAge Generator statement that the program processes. You must use the `/TRACE=STMT` generation option to enable this type of tracing. The default for the `APPSTMT` option is `N`.

- `SQLIO=Y|N`

If `SQLIO=Y` is specified, the trace facility provides traces of the data and error codes returned after the SQL call. The VisualAge Generator SQL process name, the process option, the process object, the SQL function name, and the VisualAge Generator data item name, length, type, and contents are given. You must use the `/TRACE=SQLIO` generation option to enable this type of tracing. The default for the `SQLIO` option is `N`.

- `SQLERR=Y|N`

If `SQLERR=Y` is specified, the trace facility provides traces of the error information that comes back from DB2/VSE on every database call. The `SQLCODE`, `SQLERRP`, `SQLSTATE`, `SQLWARN`, `SQLERRD`, `SQLTEXT`, and `SQLERRMC` codes are given. You must

use the /TRACE=SQLERR or the /TRACE=SQLIO generation option to enable this type of tracing. The default for the SQLERR option is N.

- IDUMP=Y | N

If IDUMP=Y is specified, the trace facility provides dumps of certain VisualAge Generator Server internal storage areas. The default for the IDUMP option is N, no internal storage dumps.

:APPLS

Application program names or print service program names

If program names are specified, only the specified programs are traced. Otherwise service requests from each specified program are traced. Up to 16 program names can be specified.

:SERVICES

Service numbers

If service numbers are specified, only those specific services are traced. To trace all service numbers, 999 must be specified. Otherwise, up to 32 service numbers can be specified.

Note: The entry to ELAZSINI (initialization service) and the exit from ELAZFTRM (cleanup service) are not traced. ELAZSINI initializes the trace facility. ELAZFTRM ends the trace facility.

If you are using the trace facility to aid in problem determination, set the control cards as directed by the IBM Support Center.

Deactivating a Trace Session

To stop all trace activity for VSE CICS transactions, use the ELAZ transaction to delete the transaction codes from the list of transactions to be traced by using the F10 function key. When a transaction ends and is subsequently restarted, tracing does not start if the transaction code no longer appears in the transaction list.

To stop tracing in VSE batch, cancel the program and remove the trace filters from the runtime JCL.

Printing the Trace Output

If trace output is directed to a file, the trace output is sent to a VSE/POWER LST queue. Use your normal method for viewing or printing a VSE/POWER LST queue member.

If trace output is not directed to a file, trace output is kept in a wrap-around buffer in memory. The trace output can be reviewed in dumps taken when programs end abnormally.

VSE CICS

The output of the trace is spooled to a VSE/POWER LST queue if you direct the trace output to a file by specifying yes (Y) on the ELAZ02 panel. If you choose the defaults, the CICS Report Controller can be used to print the trace output.

VSE Batch

The output of the trace is spooled to a VSE/POWER LST queue with the job name of ELATOUT.

Reporting Problems for VisualAge Generator Server

VisualAge Generator provides a set of manuals that contain information that can assist you in using the VisualAge Generator products. Check these manuals first for answers to your problems or questions. A list of the VisualAge Generator manuals are located in the Preface section of each manual.

Documenting information is vital not only in locating the source of the problem, but also in resolving the problem.

Gather the necessary documented information before you contact the IBM Support Center to report your problem. The following list outlines the minimum required documentation.

- The release level of VisualAge Generator Server and the operating system it is running under.
- The list of all PTFs and additional APARs that have been applied to VisualAge Generator Server
- All VisualAge Generator Server and non-VisualAge Generator Server error messages, including text and return codes
- The first indication that the problem has occurred
- A detailed description of the problem and any additional information attributing to the problem
- Determine whether the problem occurred during generation, compilation, or runtime time
- Can the problem be created again?
- Does the program run successfully under VisualAge Generator Developer?

- Do other VisualAge Generator Server generated programs end with errors? If not, does the problem program do something unique that the programs that work do not do?
- Does the program interface with non-VisualAge Generator programs? If so, what language are the non-VisualAge Generator programs written in?
- Any recent changes made to VisualAge Generator Server, the program, VSE, DB2/VSE, or other related software
- The terminal types involved if the problem is terminal related
- Languages installed and the one used to create the problem again
- For the VSE CICS environment:
 - Is the program to run as segmented or nonsegmented?
 - What diagnostic controller options are being used?

Your IBM Support Center representative might request documentation depending on the type of problem. This documentation might include the following:

- The program with associates exported in external source format
- The generated COBOL source code
- The generation options
- JCL output from preparation or runtime
- DB2/VSE information (table definitions, EXPLAIN output)
- CICS table entry information (PCT, PPT, FCT, and so forth)
- Formatted dump of the problem

These files can be put on tape or cartridge and sent to the support center. Please follow these instructions when creating the tape or cartridge:

1. Create all tapes (reels or cartridges) as unlabeled tapes.
2. Use ONLY IBM standard utilities, such as LIBRARIAN, DITTO, IDCAMS, or VSE/POWER POFFLOAD.
3. Provide a hard copy of the JCL used to create the tape.
4. Any data that you supply should be marked with your:
 - Account name
 - Problem number
 - APAR number (if assigned)

Problem Reporting Sequence

When you report a problem, tell the IBM Support Center representative the symptom string you received if an ABEND occurred. A symptom string contains an ABEND code, the name of the VisualAge Generator Server module in control at the time of the problem, and the program ID.

Figure 80 shows the sequence of events that should occur after you describe the problem to the IBM Support Center.

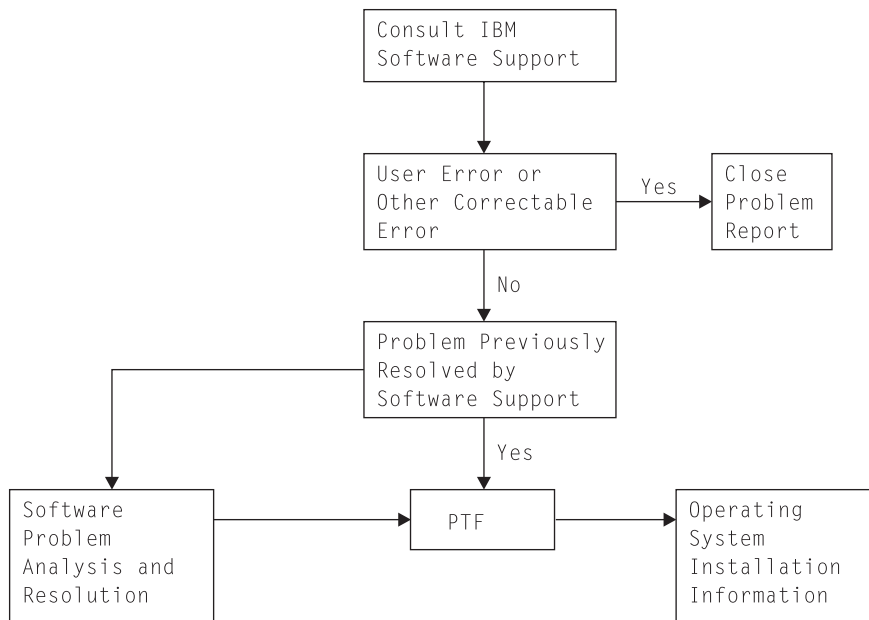


Figure 80. Problem Reporting Sequence

If the IBM Support Center has an existing solution for the problem, see the appropriate system documentation for instructions on applying the patch (if the resolution is a temporary patch) or program temporary fix (PTF).

If the IBM Support Center cannot resolve the problem, or a PTF is unavailable, an Authorized Program Analysis Report (APAR) number will be assigned to your problem. You might be asked to assist in collecting information to submit an APAR.

Chapter 43. Common Messages during Preparation for VSE Systems

This chapter contains some error messages from other products. It is not a complete list. For a complete explanation of product messages, refer to the documentation provided with that product.

VSE System and Job Control Messages

Only the most frequently occurring VSE system and job control messages are listed in this section. If you receive other messages that start with IES or 1S, or if you need a more complete explanation of one of the messages, refer to the messages and codes manual for your release of VSE/ESA.

IESI0230I MEMBER XXX.PROC DOES NOT EXIST IN LIBRARY SSSS.LLLLLL PROCESSING CONTINUES, IN CASE OF NO IESI0230I(+) ERRORS OCCUR THE MEMBER WILL BE CREATED.

Explanation: The specified PROC member used in the error processing for the preparation job does not exist. These are normal messages. Refer to “Error Processing for Preparation Jobs” on page 343 for more detailed information.

User Response: These are normal messages that you can ignore.

1S78I JOB TERMINATED DUE TO 'GOTO SEOJ' At the end of the preparation, the procedure used for error processing in the previous job is deleted. After deleting this procedure, the job issues this message and immediately terminate. Please refer to “Error Processing for Preparation Jobs” on page 343 for more detailed information.

User Response: Examine the job output that contains this message and take the proper action.

Workstation File Transfer Messages

Only the most frequently occurring file transfer message is listed in this section. If you receive other messages that start with INW or if you need a more complete explanation of one of the messages, refer to the messages and codes manual for your release of VSE/ESA.

INW0092I LIBRARY, library name IS FULL

Explanation: There is not enough space left in the library to store the new member. The file transfer is terminated.

User Response: Do one of the following:

- If the indicated library is shared between partitions on a single VSE/ESA system or that is shared between different VSE/ESA systems, submit the sample job shown in Figure 53 on page 298 to release the space for reuse.
- Extend the library as described in the VSE/ESA administration manual.

NX0002I **Transmission error.**
Module=nnnnnnnn RC=xxxx

Explanation: An error condition was detected during the data transfer between host and PC. This may be an error in the data being transferred, or an unidentified system error. nnnnnnnn is the name of the module that detected the error. xxxx is a hexadecimal code that further identifies the error.

If you work with a PC in DFT mode, and this message appears in the host session after a SEND or RECEIVE command with a module name that has a zero as last character (for example, INWPPUT0 or INWPGET0), then your CICS terminal probably does not have the EXTDS feature defined in the terminal control table (TCT).

If you work with the IBM PC3270 Emulation program, MODULE=INWPUT1 RC=0200 indicates that the PC file was not found or the diskette drive was not ready.

User Response: Examine the file for incorrect format or bad data. If your PC works in DFT mode, make sure that your CICS terminal has the EXTDS feature defined in the TCT. For further problem determination, you can add the TRACE option to your SEND or RECEIVE command. Adding the TRACE option generates an entry in the VSE/POWER queue called IWSTRACE that is likely to contain useful debugging information. Refer to the *VSE/ESA Programming and Workstation Guide* for additional information.

DB2/VSE Precompiler Messages

Only the most frequently occurring DB2/VSE precompiler messages are listed in this section. If you receive other messages that start with ARI or if you need a more complete explanation of one of the messages, refer to the documentation for your release of DB2/VSE.

ARI0501I **AN SQL WARNING HAS OCCURRED. SQLCODE DESCRIPTIVE TEXT.**

Explanation: This message is followed by message ARI505I, which provides the SQLCODE and ROWCOUNT information. The SQLCODE descriptive text describes the SQLCODE error condition. If SQLCODE is 204, one or more SQL tables have not been created. The tables that do not exist are identified in an explanation associated with the message by:

xxxxxxx was not found in the system catalog where xxxxxxxx is the table name.

User Response: Create the necessary SQL tables and synonyms before the running the program.

ARI0554E **SQL STATEMENT CONTAINS MORE THAN 256 HOST VARIABLES.**

Explanation: A limit of the host variable reference in a single SQL statement has been exceeded.

User Response: Change the SQL statement to contain fewer than 256 host variables.

ARI0565I **THE INPUT SOURCE CONTAINS EITHER NO SQL STATEMENTS, OR NO SQL STATEMENTS THAT WILL CREATE A PACKAGE.**

Explanation: This can occur if one of the following is true:

- The program contains no SQL process options.
- The program contains only CALL EZECONCT, EZEROLLB, RESET or EZECOMIT statements.

These calls result in CONNECT, ROLLBACK or COMMIT SQL statements that do not create a package.

User Response: The preprocessing continues. The generated program can be run successfully. If you have SQL process options that result in SQL calls other than CONNECT, ROLLBACK, or COMMIT in your program, then you should not receive this message.

ARI5387E The SQL statement length exceeds 8192 positions

Explanation: A limit of the SQL statement length has been exceeded. The possible causes are the following:

- The SQL statement is too long.
- The end of statement delimiter is missing.
- A FOR UPDATE OF clause is added to the end of a static query statement.

- The expansion of a host structure reference has increased the size of the SQL statement.

User Response: Either break the SQL statement into smaller statements or remove unnecessary conditions that might exist in the statement.

ARI5400I Warning: statement-type is not supported.

Explanation: The SQL statement shown is not supported by the DB2/VSE system. The statement-type can be either DECLARE TABLE, DECLARE STATEMENT, or DISCONNECT.

User Response: If your program depends on the code expansion, change your program so that it does not depend on either of the above statements. Or, install the release of DB2/VSE that supports this statement-type.

COBOL Compilation Messages

Only the most frequently occurring COBOL compilation messages are listed in this section. If you receive other compilation messages that start with IGY or if you need a more complete explanation of one of the messages, refer to the documentation for your release of COBOL.

IGYGR1471W

The system-name specified in the "ASSIGN" clause for file "EZEFILE-EZEPRINT" was longer than seven characters. Under VSE the first seven characters will be used at execution time.

Explanation: If your program displays printer maps on VSE batch, you might receive this COBOL informational message in your COBOL compilation listing.

User Response: None required. You can ignore these messages; they are normal for a program that displays printer maps on VSE batch.

IGYOP3091W

Code from "?" to "?" can never be executed, and was therefore discarded.

IGYOP3093W

The "PERFORM" statement at "?" cannot reach its exit.

IGYOP3094W

There may be a loop from the "PERFORM" statement at "?" to itself. "PERFORM" statement optimization was not attempted.

Explanation: These messages occur if your program has been processed using the OPTIMIZE compiler option.

User Response: These are normal messages that you can ignore.

IGYPA3013W

Data item "?" and "?" had overlapping storage. An overlapping move will occur at execution time.

Explanation: This message occurs if your program attempts to assign the value of a data item to the same data item.

User Response: You might want to check that you really intended to do this.

IGYPG3113W

Truncation of high-order digit positions may occur due to precision of intermediate results exceeding 30.

Explanation: This message might occur if your program was generated with the /MATH=COBOL generation option.

User Response: You might want to change the arithmetic expression identified in the message. For example, you could split the expression into several smaller ones.

If you do not change the expression, ensure that the intermediate values will fall within the precision that COBOL supports. Refer to the application programming guide for your release of COBOL for more information about the precision of intermediate results.

IGYPS2015I The paragraph or section prior to paragraph or section EZEMAIN-PROCESS did not contain any statements.

IGYPS2023I Paragraphs prior to section EZEMAIN-PROCESS were not contained in a section.

Explanation: These two messages occur if your program has been processed by the DB2/VSE precompiler.

User Response: They are normal messages that you can ignore.

IGYSC2025W

"EZEPCB-?" or one of its subordinates was referenced, but "EZEPCB-?" was a "LINKAGE SECTION" item that did not have addressability. This reference will not be resolved successfully at execution.

Explanation: This warning message occurs when PCBs or any data structure is generated in the linkage section, but is not used in a statement.

User Response: Ignore the messages and the program will work correctly.

Chapter 44. Common System Return Codes for VSE Systems

VisualAge Generator Server messages might include return codes from databases for operating systems that are being used. This could include DL/I, DB2/VSE, VSE CICS, or VSE batch return codes.

This chapter contains only the most common errors that occur during file input and output operations.

The return codes included in this chapter are for the following databases and operating systems:

- VSE CICS
- COBOL
- DL/I
- DB2/VSE
- VSAM
- VSE batch

Common EZERT8 Return Codes

The EZERT8 special function word contains a return code indicating a reason that a file I/O process option or a CALL to a service routine is not successful. Codes specific to the system or the access method are returned when the user requests system codes with the /SYSCODES generation option.

If the /NOSYSCODES generation option is specified, the system return codes are converted to VisualAge Generator Server return codes.

System Return Code Formats for EZERT8

The following table shows the formats of EZERT8 by specific environment:

System	Compatibility Considerations
VSE CICS	<p>If EZERT8 is in the form RSnnnnnn, look under nnnnnn in “Common System Return Codes in EZERT8” on page 433.</p> <p>Otherwise, the first 2 characters of EZERT8 contain the first byte of the EIBFN from the CICS EXEC interface block (EIB). The remaining 6 characters contain bytes 0-2 of the EIBRCODE, also from the CICS EXEC interface block (EIB).</p> <p>If the REPLY option is specified on a CALL statement, and the CALL is implemented with the VSE CICS LINK, the first 2 characters of EZERT8 contains the first byte of the EIBFN and the remaining 6 characters contain bytes 0-2 of the EIBRCODE.</p> <p>For an explanation of the EIB codes, refer to the <i>VSE CICS Application Programmer's Reference</i></p> <p>For files accessed through COBOL, EZERT8 contains file status key 1 in the first character and file status key 2 in the next 3 characters, followed by 4 characters of zeros. If file status key 1 is the character '9', file status key 2 is converted from single-byte binary to 3-character decimal format.</p> <p>For files accessed through VSE CICS, the first 2 characters of EZERT8 contain the first byte of the EIBFN field from the CICS EXEC interface block (EIB). The remaining 6 characters contain bytes 0-2 of the EIBRCODE, also from the CICS EXEC interface block (EIB).</p> <p>For SPOOL files accessed through VSE CICS, EZERT8 contains the return code from VSE/POWER. The first two characters of EZERT8 contain the hexadecimal representation of the first byte of the EIBFN from the CICS EXEC interface block. The third and fourth characters of EZERT8 contain byte 4 of the EIBRCODE, also from the CICS EXEC interface block. The remaining 4 characters of EZERT8 contain the hexadecimal representation of the EIBRESP2 code from the CICS EXEC interface block. EIBRESP2 contains the actual VSE/POWER return and feedback codes.</p>

System	Compatibility Considerations
VSE batch	<p>If EZERT8 is in the form RSnnnnnn, look under nnnnnn in “Common System Return Codes in EZERT8”. The EZERT8 special function word contains a values that depend upon the file type as follows:::</p> <p>SEQ: EZERT8 contains the COBOL status key value or values in the first 2 characters. The remaining 6 characters are zeros.</p> <p>VSAM: EZERT8 contains the COBOL status key value or values in the first 2 characters followed by 2 characters for the COBOL VSAM return code (VSAM feedback code), 1 character for the COBOL VSAM function code (VSAM component code), and 3 characters for the COBOL VSAM feedback code (VSAM reason code).</p> <p>VSAMRS: The contents of EZERT8 depend on the operation that was not successful:</p> <ul style="list-style-type: none"> • If an OPEN or CLOSE is not successful, the first 2 bytes of EZERT8 contain the error code from the VSAM application control block (ACB) in hexadecimal. The remaining 6 characters are zeros. • If an operation other than OPEN or CLOSE is not successful, the first 2 characters are zeros followed by 2 characters for the COBOL VSAM return code (VSAM feedback code), 1 character for the COBOL VSAM function code (VSAM component code), and 3 characters for the COBOL VSAM feedback code (VSAM reason code). <p>SPOOL: The first two characters of EZERT8 contain the return code from the XPCCB (the value of IJBXRETC). The 3rd and 4th characters of EZERT8 contain the return code from VSE/POWER (the value of PXPRETCD). The 5th and 6th characters of EZERT8 contain the feedback code from VSE/POWER (the value of PXPFBKCD). The remaining two characters contain zeros.</p>

Common System Return Codes in EZERT8

The following table gives an explanation of the most common values that you receive in EZERT8 when system codes are requested. If your return code is not listed here, or you would like more information, see the table in “System Return Code Formats for EZERT8” on page 432 and then the appropriate manuals for your environment.

Table 62. EZERT8 Return Codes.

System	Return code	Meaning
VSE batch	6E000000	VSAM open error - empty file opened for input
VSE batch	40000000	VSAM open error - file is not in CI format

Table 62. EZERT8 Return Codes. (continued)

System	Return code	Meaning
VSE batch	80000000	VSAM open error - DLBL is missing or file name does not match ACB
VSE batch	00000004 on non-VSAM file	End of file
VSE batch	00000008 on non-VSAM file	Error opening file
VSE batch	00000012 on non-VSAM file	Error on I/O or closing a file
VSE batch	0008*004 for sequential retrieval	End of file
VSE batch	0008*004 for relative	No record found
VSE batch	0008*008 for an ADD request	Duplicate record
VSE batch	0008*016	No record found
VSE batch	0008*016 for relative	Deleted or invalid record detected.
VSE batch	0008*028	File full
VSE CICS	ffrrrrr Remote CALL or CREATX	Other VSE CICS errors where: <ul style="list-style-type: none"> • ff = Hexadecimal representation of EIBFN byte 0 • rrrrrr = Hexadecimal representation of EIBRCODE bytes 0-2
VSE CICS	0A010000 on scanning a temporary storage queue	End of file
VSE CICS	0A010000 on direct I/O to a temporary storage queue	No record found
VSE CICS	0A080000 on temporary storage queue	File is full
VSE CICS	060F0000 on VSAM file	End of file
VSE CICS	00000000 Remote CALL or CREATX	Successful
VSE CICS	00000203 Remote CREATX	Transaction identifier is not valid
VSE CICS	00000204 Remote CALL	Program name is not valid
VSE CICS	00000207 Remote CALL or CREATX	System identifier is not valid
VSE CICS	00000208 Remote CALL	Link is out of service
VSE CICS	06810000 on VSAM file	No record found
VSE CICS	06820000 on VSAM file	Duplicate record
VSE CICS	06830000 on VSAM file	File is full
VSE CICS	08E10000 on transient data	Format error
VSE CICS	08010000 on a transient data queue	End of file
VSE CICS	08020000 on a transient data queue	File not found
VSE CICS	08080000 on transient data	Transient data queue not open
VSE CICS	12320000	Queue is already in use

VisualAge Generator Server Return Codes

The return codes list is sequenced by return code, with the alphabetic return codes (A to Z) occurring before the numeric return codes (0 to 9).

Table 63. VisualAge Generator Server Return Codes

RC	Component	Probable Cause
Cnn	File control/request	These return codes do not have a VisualAge Generator mnemonic. Either MVS CICS or VSE CICS returned an IOERR error (if MVS CICS or VSE CICS), or VSAM returned a return code of 12 on file input/output. The nn is replaced by the VSAM reason code from the feedback field. For more information, see the <i>VSE/ESA Messages and Codes</i> manual.
Fnn	File control/request	These return codes are VSE CICS EIBRCODES, other than ILLOGIC, IOERR, and those that have VisualAge Generator mnemonics. The nn is replaced by the EIBRCODE (byte 0). For more information, see the <i>CICS/VS Application Programmer's Reference Manual (Command Level)</i> Note: All return codes, other than the ones that have VisualAge Generator mnemonics, cause the program to end. An error message is issued to inform you that the program ended because of a send/receive error. The error message includes the return code.
FE1	File Control/request	Transient data queue - Queue length and VisualAge Generator record length do not match. The VisualAge Generator mnemonic FMT is set.
F02	File Control/request	Transient data queue - File not found. VisualAge Generator mnemonic FNF is set.
F08	File control/request	An attempt was made to access an extrapartition transient data queue, but the queue has not been opened yet. Exit and use CEMT to open the queue.
02	EZCHART	An incorrect number of parameters was passed on a call to EZCHART.

Table 63. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
03	EZCHART	GETMAIN error - An attempt to obtain storage for data conversions was not successful.
04	EZCHART	A data overflow occurred during the conversion of numeric data to floating point format. Data must fit into 6-place floating point format.
06	EZCHART	The printer depth was not specified in Chart Control.
07	EZCHART	The printer width was not specified in Chart Control.
08	EZCHART	The number of print copies was not specified in Chart Control.
09	EZCHART	The printer name was not specified in Chart Control.
10	EZCHART	A load or link error occurred for a GDDM* routine.
52	Terminal support	You attempted to enter VisualAge Generator on a 3278-52 terminal device. This device is not supported by VisualAge Generator.
101	Message processing	The message was truncated.
102	Contents control	The module specified on a LOAD macro is already in storage.
	File control/request	The end of file was reached. The VisualAge Generator mnemonic EOF is set. Note: EOF is set when a SCAN is performed on an empty file.
	Service request	An ITEMERR condition was received from MVS CICS or VSE CICS because the maximum number of records allowed in a temporary storage queue (32767) was exceeded.
103	File control/request	You performed an operation on a record that has a duplicate key, or a key in the record for an alternate index is duplicated. The I/O operation completed, and the VisualAge Generator mnemonic DUP is set.

Table 63. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
104	File control/request	The end of file was reached. The VisualAge Generator mnemonic EOF is set.
115	Service request	An EXEC CICS ENQ was not successful.
116	Service request	An EXEC CICS DEQ was not successful.
20B	Storage allocation	Operands that are not valid were specified on either a GETMAIN or FREEMAIN macro.
20C	Defined data set	The data set name specified on an issued DEFDS command already exists in the external work file.
	Storage allocation	An error occurred while processing a FREEMAIN macro.
200	Service request	An service request was issued that is not valid. This is a system error. Contact the IBM Support Center.
201	File open/connect	The connection already exists. The possible cause is a file with the same name is already in use. VisualAge Generator mnemonic FNA is set.
	Message processing	Variables were passed to be built into the message, but the message contained no variable fields; or, the message contained variable fields, and no variables were passed for them.
201- 206	Service request	Service request errors occurred while processing a DEFDS command. This is a system error. Contact the IBM Support Center.
202	File control/request	Record not found. The VisualAge Generator mnemonic NRF is set.
	Storage allocation	The ORIGIN specified on a FREEMAIN macro does not match storage already in use.
203	File control/request	The record was not found. The VisualAge Generator mnemonic NRF is set.

Table 63. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
	Storage allocation	Either the ORIGIN specified on a FREEMAIN macro does not begin on a doubleword boundary, or 0 LENGTH was specified on a GETMAIN.
204	Storage allocation	An attempt has been made to free storage that has not been allocated or that has already been freed.
205	File control/request	The record was not found. The VisualAge Generator mnemonic NRF is set. Note: The VisualAge Generator mnemonic NRF is set when a SCAN or SCANBACK is preceded by a SET record SCAN on an empty indexed file.
	Storage allocation	The LENGTH specified on a FREEMAIN macro is 0.
206	File control/request	You attempted to store a record with a duplicate key while using an index that does not allow duplicate keys. The VisualAge Generator mnemonics UNQ and DUP are set.
207	File control/request	The record was not found. The VisualAge Generator mnemonic NRF is set.
208	File control/request	An error occurred when you attempted to connect or write to the log file on MVS CICS or VSE CICS. A possible reason for the error is that no entry was found for the log file in the destination control table (DCT).
	Service request	The NDSNAME in an ALTDS request is not valid. This is a system error. Contact the IBM Support Center.
	Storage allocation	The storage specified on a FREEMAIN macro is already free.
209	Service request	The name specified by the NDSNAME on an ALTDS command already exists in the external work file. This is a system error. Contact the IBM Support Center.

Table 63. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
210- 211	Service request	Miscellaneous errors occurred on an ALTDS request. This is a system error. Contact the IBM Support Center.
212	Service request	An I/O error occurred while copying data from the work file to the external work file during an ALTDS service request.
213	Service request	The COPIES operand on a SUBMIT.PRINT service request is not valid. This is a system error. Contact the IBM Support Center.
214	Service request	The data set on a SUBMIT.PRINT service request cannot be found. This is a system error. Contact the IBM Support Center.
215	File control/request	You attempted to store a record with a duplicate key while using an index that does not allow duplicate keys. The VisualAge Generator mnemonics UNQ and DUP are set.
216	File open/connect	A connection was attempted to an ESDS file or transient data in direct mode. The VisualAge Generator mnemonic FMT is set.
	Service request	The data set specified on a DEFDS request matches a CONNECT already in use. This is a system error. Contact the IBM Support Center.
217	File open/connect	An attempt was made to subconnect to a serial file. Check to see if a called program is attempting to reference the same serial file that has been referenced by the calling program.

Table 63. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
	Service request	<p>A PRINT error has occurred for one of the following reasons:</p> <ul style="list-style-type: none"> An error occurred while writing to the transient data queue on MVS CICS or VSE CICS. The most common errors are QIDERR, IOERR, LENGERR, and NOSPACE. An error occurred while writing to the data definition name (DD name) EZEPRINT in MVS/TSO or MVS Batch or to the disk label (DLBL) EZEPRIN in VSE batch. A possible cause is that the printer file (for example, EZEPRINT) has been allocated incorrectly or not been allocated at all. On MVS systems you get this error, for example, if the data set allocated for the print output has the wrong record format (anything other than VBA) or the wrong record length (shorter than the print output line length). You get this error if the allocation statement (in the CLIST or startup JCL) is missing or incorrect.
218	Service request	The file is not available. The VisualAge Generator mnemonic FNA is set.
22A	File control/request	The available storage space has been exhausted. Try the operation again. If the problem persists, contact your system programmer.
220	File open/connect	<p>A format error occurred. Either the characteristics of a file are not supported by VisualAge Generator Developer, or they are incompatible with the VisualAge Generator Developer record. The VisualAge Generator mnemonic FMT is set.</p> <p>In MVS/TSO, this occurs if no member name was specified for accessing a PDS data set.</p>
	File control/request	The record length for a file is larger than the maximum record length defined in the system.

Table 63. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
221	Service request	An ENQ was not successful while writing to the transient data queue on MVS CICS. This is a system error. Contact the IBM Support Center.
223	Service request	The attach of the print subtask was not successful, or the print subtask abended. This is a system error. Contact the IBM Support Center.
225	Service request	The print subtask abended. This is a system error. Contact the IBM Support Center.
226	File control/request	An IO error occurred while reading or writing from temporary storage on MVS CICS or VSE CICS. This is a system error. Contact the IBM Support Center.
25A	File control/request	The data set cannot be extended because VSAM cannot allocate additional direct-access storage space. Either not enough space is left to make the secondary allocation request, or you attempted to increase the size of a data set while processing with SHROPT=4 and DISP=SHR. The VisualAge Generator mnemonic FUL is set.
251	File open/connect	For MVS CICS or VSE CICS environments, the file control table (FCT) entry was not found, indicating a real file or transient data queue was not properly defined or generated. For MVS/TSO environments, either the ddname has not been allocated, or the data set for the dynamic allocation does not exist.
280	File control/request	The data set that you are trying to connect to is already in use. A probable cause is that your program has a data set associated to one record and you are trying to use another record with the same data set associated. You need to issue a CLOSE on the first record to free the data set before trying to use it with another record.
291– 294	Service request	A mapping error occurred.

Table 63. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
	Terminal support	A mapping error occurred.
380	File control/request	A deadlock occurred. One transaction is attempting to update a record that is currently locked by another transaction. The VisualAge Generator mnemonic DED is set.
381	File control/request	The control interval for a record is already held in exclusive control by another requester. The VisualAge Generator mnemonic LOC is set. For MVS CICS or VSE CICS systems, the returned MVS CICS or VSE CICS return code is INVREQ and this is assumed to have occurred due to one transaction's attempt to do two UPDATES to the same file. If this is not the case, see the description of INVREQ in the <i>CICS Application Programmer's Reference</i> manual.
389	File control/request	The resource control block could not be found to process the request against. This is a system error. Contact the IBM Support Center.
399	File control/request	You attempted to store a record to a temporary storage queue with a key that exceeds 32767. The key is too large for temporary storage queues, which cannot have more than 32767 records.
4nn	File open/connect	For MVS/TSO systems only, the VSAM GENCB for an ACB was not successful. The value of nn is determined from VSAM return codes. If register 15 contains 4, nn is replaced by the contents of register 0. If register 15 does not contain 4 (or 0), nn is replaced by 50 plus the contents of register 15.

Table 63. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
5nn	File open/connect problem, abend, end, was not successful; not fail or fails CLT 1/11/94	For MVS/TSO systems, an OPEN request is not successful. For VSAM files on MVS/TSO, a SHOWCB for the ERROR field is done after the problem with the OPEN request. The value of the ERROR field replaces nn . For non-VSAM sequential files in the MVS/TSO environment (QSAM), nn is replaced with a value of 0. For spool files in a CICS environment, nn is also replaced with 0.
5A0	File open/connect	An attempt was made to open a VSAM data set for input, but the data set was empty.
6nn	File open/connect	The VSAM GENCB for an RPL was not successful. For MVS/TSO, the value of nn is determined from VSAM return codes. If register 15 contains 4, nn is replaced by the contents of register 0. If register 15 does not contain 4 (or 0), nn is replaced by 50 plus the contents of register 15.
701	File open/connect	On VSE CICS 2.1 or later, the file cannot be opened or connected. The error is not defined in the FCT flags.
702	File open/connect	The VSAM SHOWCB or MODCB macro was not successful. This usually means that the file is not open.
703	File open/connect	The VSAM TESTCB macro was not successful.
705	File open/connect	For VSE CICS only, a connection was attempted to transient data or a temporary storage queue, but a VSAM file has the same name.
706	File open/connect	On VSE CICS 2.1 or later, the file is UNENABLED and cannot be opened or connected.
707	File open/connect	On VSE CICS 2.1 or later, the file is DISABLED or DISABLING and cannot be opened or connected.
708	File open/connect	On VSE CICS 2.1 or later, the user is not authorized to access the file.

Table 63. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
709	File open/connect	On VSE CICS 2.1 or later, an I/O error occurred on the SET data set Open command.
768	File open/connect	The OPEN or connection was not successful due to a problem with a GETMAIN when requesting storage for control blocks associated with sequential files.
8nn	File control/request	These return codes do not have a VisualAge Generator mnemonic. Either VSE CICS returned an ILLOGIC error (if VSE CICS), or VSAM returned a return code of 8 on file input/output. The nn is replaced with the VSAM error code. For more information, see the <i>VSE VSAM administration Manual: Macro Instruction Reference</i> manual.
	File open/connect	A storage allocation was not successful.
80C	Storage allocation	There is insufficient storage to satisfy a GETMAIN request.
802	File open/connect	The resource is not associated.
	Storage allocation	There is insufficient storage for allocation.
803	Contents control	The module specified on a LOAD macro could not be located.
804	File control/request	This return code is received from MVS CICS or VSE CICS and indicates that a QIDERR or ITEMERR occurred while trying to access a temporary storage queue.
805	Contents control	The module specified on a LOAD macro is damaged.
	Message processing	An unsupported option was specified on an INFORM macro.
806	Contents control	For MVS/TSO, the module specified on a LOAD macro could not be located.
807	Contents control	Insufficient storage is available to load the specified module.
81C	File Control/ Request	A Temporary Storage queue is full. The VisualAge Generator mnemonic FUL is set.

Table 63. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
988	File control/request	For MVS/TSO only, a branch was made to the SYNAD routine as the result of a PUT to a non-VSAM serial file. A possible reason is that the file is empty or the blocking factor is wrong. For MVS CICS or VSE CICS, a WRITE request to a spool file was not successful.
989	File control/request	An error occurred while trying to close a file.
999	File control/request	An unsupported request was made for a serial file. A probable cause is that the VisualAge Generator Developer record associated with this file was not defined with a serial organization.

Common DB2/VSE Return Codes

Only the most frequently occurring DB2/VSE codes are listed in this section. If you receive other DB2/VSE codes or if you need a more complete explanation of one of the DB2/VSE codes, refer to the documentation for your release of DB2/VSE.

RC Meaning

100 No rows were found by DB2/VSE that meet the search criteria specified in the WHERE clause of the DB2/VSE statement, or if processing a SCAN process option in conjunction with a SETINQ or SETUPD process option, the end of the selected rows has been reached. The possible causes are the following:

- The key value(s) were not moved correctly to the host variable(s) used in the WHERE clause.
- No rows meet the search criteria specified in the WHERE clause.
- VisualAge Generator Developer stripped trailing blanks for the character host variables used in a LIKE predicate in the WHERE clause. You can use the /TRACE=SQLIO generation option to enable tracing of the data sent to DB2/VSE and the data coming back from DB2/VSE. See “Using the VisualAge Generator Server Trace Facility” on page 414 for more information about using the VisualAge Generator Server trace facility.

-301, -302, -303, -304

The VisualAge Generator Developer data item definition does not match the definition of the same column in the DB2/VSE table. This

can be caused by defining a column as variable length, but not defining the data item in VisualAge Generator Developer with a variable-length DB2/VSE code.

Make the necessary changes in the VisualAge Generator Developer data item definition to match the DB2/VSE table and generate the program again.

Refer to the *Program Directory for VisualAge Generator Server for VSE* for information about installing a DB2/VSE work database.

-805 The package *owner.package-name* is not in the SYSTEM.SYSACCESS catalog table.

The package name in the message insert area returned in the SQLCA is the name of the program that issued the SQL request. The DB2/VSE package associated with the program has not been installed in the database your program is trying to access.

-818 The DB2/VSE precompiler-generated time stamp in the phase is different from the time stamp in the package associated with the program issuing the SQL request. Both of these precompiler outputs must match and one of them is not from the most recently run precompile. This typically happens when a program used with more than one database is modified, generated again, and precompiled and the package produced by the precompilation has not been copied to all the databases that the program accesses; or, the package is installed in the database and someone tries to run the previous version of the program against the database.

Another possible cause when running on VSE CICS is that an old copy of the phase is resident in storage. It must be replaced with a VSE CICS NEWCOPY command.

Ensure that you are running with the most recent copy of the phase. On VSE CICS ensure that the latest copy of the phase has been picked up by issuing a VSE CICS NEWCOPY command or by using the VisualAge Generator Server new copy utility.

-911, -913

A deadlock condition occurred. Possible causes are:

- The isolation level was set for repeatable read.
- There were long periods of time between commit points.
- The program issued an update process without a replace option.

Note: The program should be coded to handle these conditions.

Common DL/I Status Codes

Only the most frequently occurring DL/I status codes are listed in this section. If you receive other DL/I status codes or if you need a more complete explanation of one of the DL/I status codes, refer to the application programming manual for your release of DL/I DOS/VS.

Code Meaning

- AD** The function parameter on the call is not valid. If the function code is correct, the status code can be from an I/O or alternate PCB for a database call. You might have a mismatch between the PSB you defined in VisualAge Generator Developer and the DL/I DOS/VS PSB definition.
- AI** A data management open error occurred. Either no DLBL statements were supplied for logically related databases, or DLBL name is not the same as the name specified on the DATASET statement of the DBD. The segment name area in the DB PCB has the DLBL name of the data set that could not be opened.
- AJ** The format of one of your SSAs is not valid. Either the SSA contains a command code for that call that is not valid, or the SSA qualification format is not valid.
- AK** An SSA contains either a field name that is not valid or a name that is not defined in the DBD, or the VisualAge Generator Developer data item for DL/I segment does not match the name defined to DL/I.
- AM** The call function is not compatible with the processing option in the PCB, the segment sensitivity, the transaction-code definition, or the program type.
- GA** In trying to satisfy an unqualified GN or GNP call, DL/I crossed a hierarchic boundary into a higher level.
- GB** In trying to satisfy a GN, DL/I reached the end of the database.
- GE** DL/I is unable to find a segment that satisfies the segment described in a get call.
- GK** DL/I has returned a different segment type at the same hierarchic level for an unqualified GN or GNP.
- GP** The program issued a GNP when parentage is not established, or the segment level specified in the GNP is not lower than the level of the established parent.
- II** The program issued an ISRT that tried to insert a segment that already exists in the database.

Common VSAM Status Codes

Only the most frequently occurring VSAM status codes are listed in this section. If you receive other VSAM codes or if you need a more complete explanation of one of these values, refer to the section titled "VSE/VSAM Return and Error Codes" in the *VSE/ESA Messages and Codes* manual.

OPEN request type

Code	Meaning
------	---------

- | | |
|----|---|
| 64 | Warning message: OPEN encountered an empty alternate index that is part of an upgrade set. |
| 74 | This is a warning message indicating the data set was not properly closed. Either the OPEN's implicit verify was not successful, or the user specified that OPEN's implicit verify should not be run. A previous VSAM program might have ended abnormally. The VERIFY command of Access Method Services can be used to properly close the data set. |
| 80 | The DLBL statement for this access method control block is either missing or not valid. |
| 94 | Either no record for the data set to be opened was found in the available catalog or catalogs, or an unidentified error occurred while VSAM was searching the catalog. |
| 98 | Security verification was not successful; the password specified in the access method control block for a specified level of access does not match the password in the catalog for that level of access. |
| A0 | The operands specified in the ACB or GENCB macro are inconsistent either with each other or with the information in the catalog record. |
| A8 | The data set was not available for the type of processing you specified, or an attempt was made to open a reusable data set with the reset option while another user had the data set open. |
| BC | The ACB to be opened is already being used. |
| C0 | An unusable data set was opened for output. |
| C4 | Access to data was requested using an empty alternative index. |

CLOSE request type

Code	Meaning
------	---------

- | | |
|----|--|
| 04 | The data set indicated by the access method control block is already closed. |
|----|--|

- 88** Not enough virtual storage was available in your program's address space for the work area required by CLOSE.

GET/PUT/POINT/ERASE/CHECK/ENDREQ request types

The following occur when register 15=8(8).

Code	Meaning
-------------	----------------

- | | |
|-----------|--|
| 08 | An attempt is made to store a record with a duplicate key, or there is a duplicate record for an alternate index with the unique key option. |
| 6C | The RECLLEN specified was one of the following: <ul style="list-style-type: none">• Larger than the maximum allowed• Equal to 0• Smaller than the sum of the length and the displacement of the key field• Not equal to the record (slot) size specified for a relative record data set |
| 70 | The KEYLEN specified was too large or equal to 0. |
| 84 | An attempt was made (in locate mode) to retrieve a spanned record. The spanned record is not supported by VisualAge Generator Developer. |
| C0 | A relative record number that is not valid was encountered. |

COBOL Status Key Values

This shows the most frequently occurring COBOL status key values. If you receive other status key values or if you need a more complete explanation for one of these values, refer to the application programming language reference for your release of COBOL.

Status Key	Explanation
-------------------	--------------------

- | | |
|-----------|--|
| 10 | The end of a file was reached. |
| 22 | An attempt was made to write a record with a key that duplicated one that was already in the file. |
| 23 | Record not found. This can also be caused by an optional file not being allocated. |
| 35 | No DLBL statement was included in the JCL. This can occur if the program calls another program or transfers to another program using |

a DXFR statement but the DLBL statements for the second program have not been added to the sample runtime JCL for the main program.

- 39** The physical file that you specified during resource association does not match the file characteristics that you specified during record definition. The file characteristics include file organization (sequential, relative or indexed), the prime record key, the alternate record keys, and the maximum record size.
- 44** A variable-length record was written that is not valid. This can occur if the value in the number of occurrences field is larger than the maximum value, or the value in the record length item field is larger than the maximum length of the record.
- 96** No DLBL statement was included in the JCL for a VSAM file. This can occur if the program calls another program or transfers to another program using a DXFR statement, but the DLBL statements for the second program have not been added to the sample runtime JCL for the main program.

Chapter 45. VisualAge Generator Server Return Codes and Abend Codes for VSE Systems

The information within this chapter is diagnosis, modification, or tuning information.

Return Codes

This section contains a listing of codes set by VisualAge Generator Server and returned in the COBOL return code of a program. If a generated program completes normally, the COBOL return code is set to the value in the EZERCODE special function word. This code must be less or equal to 512. Return codes greater than 512 are reserved for VisualAge Generator Server. The return codes set by VisualAge Generator Server are:

- 693** The program ended due to an error detected by VisualAge Generator Server. The error description is reported as described in Chapter 42. Diagnosing Problems for VisualAge Generator Server on VSE Systems.

ABEND Codes

VisualAge Generator Server reports errors by error messages whenever possible. ABENDs are issued only in situations where initialization has not progressed to the point where messages can be issued or when the error messages cannot be written to their normal destination.

VSE CICS and VSE Batch Environments

For VSE CICS and VSE batch environments, you can control whether or not a dump is taken by using the diagnostic controller utility. If a dump is taken, the dump code is ELAD. See "Controlling Error Reporting in VSE CICS" on page 405 for information on the diagnostic controller utility.

- ELA1** This abend code should never be received. However, if register 1 in a dump contains "ELA1", then a database manager or subsystem interface module (for example, ASMTDLI for DL/I access) was not linked with a VisualAge Generator Server program at product installation. Registers 3 and 4 in the dump usually contains the name of the stub program. The abending phase is the phase that was not linked correctly.
- ELA2** The Task Work Area (TWA) does not exist or is not long enough to be used by VisualAge Generator Server. The TWA length must be greater

than or equal to the sum of 1024 plus the /TWAOFF (TWA offset) generation option specified when the initial program in the transaction was generated.

Use the TWASIZE parameter in the PCT entry to define a TWA with an adequate length for the transaction.

When this is a client/server program set, ensure that the CICS mirror transactions on the server CICS have a TWA size at least equal to the TWA size specified for the VisualAge Generator transaction on the client system.

ELA3 Load for module ELARSCNT was not successful. VisualAge Generator Server has not been installed correctly.

Ensure that the VSE CICS or VSE batch partition has access to the VisualAge Generator Server runtime library and that module ELARSCNT is defined in the PPT.

When this is a client/server program set, ensure that the security level-checking parameters (RSL and RSLC) for the CICS mirror transactions on the server CICS are the same as those specified for the VisualAge Generator transaction on the client system.

ELA4 Load for module ELARPRTRX was not successful. VisualAge Generator Server has not been installed correctly.

Ensure that the VSE CICS or VSE batch partition has access to the VisualAge Generator Server runtime library and that module ELARPRTRX is defined in the PPT.

ELA5 Load for module ELARPRTC was not successful. VisualAge Generator Server has not been installed correctly.

Ensure that the VSE CICS or VSE batch partition has access to the VisualAge Generator Server runtime library and that module ELARPRTC is defined in the PPT.

ELA6 The dynamic storage stack used for working storage for VisualAge Generator Server modules was exhausted and VisualAge Generator Server could not continue.

This problem should not occur. Report the problem to the IBM support center.

ELA7 A GETMAIN was not successful. There was not enough storage for the program to complete.

Try the program again when the partition is less busy or try it again in a larger partition.

ELA9 Load or link for a VisualAge Generator Server module was not successful. VisualAge Generator Server has not been installed

correctly. Use CEDF to determine the module name. Look for a PGMIDERR on a CICS LOAD or CICS LINK command.

Ensure that the VSE CICS or VSE batch partition can gain access to the VisualAge Generator Server runtime library and the module name being loaded is defined in the PPT.

ELAB A call was made to a main program, which is not allowed, or a non-VisualAge Generator program was transferred to with a DXFR statement and the NONCSP parameter was not specified either on the DXFR statement or as the LINKTYPE parameter in the linkage table.

A non-VisualAge Generator program was transferred to with a DXFR statement without identifying the target as NONCSP.

ELAC VisualAge Generator Server has detected an invalid FREEMAIN request. Collect the dump and contact the IBM Support Center for assistance.

ELAE A generated program ends abnormally because of a serious error. This occurs for one of the following reasons:

- Storage has been corrupted so that a dump is necessary to debug the abend.
- Error handling was unable to write messages to the error destination queue or to the user at the terminal. The dump is necessary to make the diagnostic information available. The situation can occur if the error destination queue specified for the transaction using the diagnostic controller utility is not defined to VSE CICS or VSE batch. In CICS, if the error destination queue is defined as an intrapartition queue, this situation occurs when there is no more space on the intrapartition queue and the error messages cannot be written.
- A severe error has occurred. Refer to the error destination queue for the corresponding error messages. The default name is ELAD. The queue name can be changed using the diagnostic controller utility.

See “VisualAge Generator Server ABEND Dumps” on page 411 for information on how to find error messages in the dump on an ELAE abend.

ELAF ELATSRST has detected one of the following errors:

- ELATSRST was not initiated with a CICS XCTL command (e.g. the restart transaction ID was associated directly to ELATSRST).
- The COMMAREA length on entry was not 0 or 10.
- The VisualAge Generator Server portion of the TWA had been initialized, indicating that a CONVERSE process option was not in

process or the non-VisualAge Generator program uses the TWA and the program was not generated with the proper TWA offset.

- Information in the COMMAREA was not valid, indicating that a CONVERSE process option was not in process.
- Information in the COMMAREA indicates that this module was started with an XFER statement with a map or during First Map processing.

ELAW A program and its associated map groups or a map group and its associated tables were generated using incompatible versions of COBOL generators. For more information, refer to the the “Transferring Program Control” section of the *VisualAge Generator Design Guide*. In addition, use the search function to locate APAR PN45782 information.

VSE Batch Environment

1600 A generated program ends abnormally because of a serious error. This occurs for one of the following reasons:

- Storage has been corrupted so that a dump is necessary to debug the abend.
- A severe error has occurred. In VSE batch, refer to SYSLST for the messages.

See “VisualAge Generator Server ABEND Dumps” on page 411 for information on how to find error messages in the dump on a 1600 abend.

1601 A database manager or subsystem interface module (for example, ASMTDLI for DL/I access) was not linked with a VisualAge Generator Server program at product installation. Registers 3 and 4 in the dump contain the name of the stub program. The abending phase is the phase that was not linked correctly.

See the *Program Directory for VisualAge Generator Server for VSE* for information on correctly linking the abending phase.

See “VisualAge Generator Server ABEND Dumps” on page 411 for information on how to find error messages in the dump on an 1602 abend. Depending on the generation options specified for the program, the message might also have been written to SYSLST. See Chapter 42. Diagnosing Problems for VisualAge Generator Server on VSE Systems for more information on VisualAge Generator Server error reporting.

1606 The dynamic storage stack used for working storage for VisualAge Generator Server modules was exhausted and VisualAge Generator Server could not continue.

This problem should not occur. Report the problem to the IBM support center.

- 1608** VisualAge Generator Server has detected an invalid FREEMAIN request. Collect the dump and contact the IBM Support Center for assistance.

Chapter 46. Codes from Other Products for VSE Systems

This chapter contains lists of common system abend codes, COBOL runtime messages and abend codes, common runtime messages from VSE CICS, VSE batch, DL/I, and SQL/DS.

Common System Abend Codes for All Environments

Only the most frequently occurring abend codes are listed in this section. If you receive another abend code or if you need a more complete explanation of one of the abend codes, refer to the VSE/ESA messages and codes manual for your release of VSE.

System 04

This code can occur on a DXFR statement if there is a mapping service or table program with the same name as the transferred-to program. This code can also occur when a mapping service or table program is called but there is a different program (for example, a non-VisualAge Generator program or an application program) with the same name. Using naming conventions can eliminate this problem.

This code can also occur if you add table edits to a map group that is shared by multiple programs and do not generate all the programs again.

System 07

Data exception. The abend occurs when fields defined as decimal or packed decimal are retrieved from a database and are found to contain data of a different format.

The abend can also occur if fields that are not initialized are used in calculations or comparisons. This happens if the program attempts to read a record from a database and the record is not found, but the program uses fields in the record anyway. To ensure that records are initialized, use a SET record EMPTY or specify /INITADDWS and /INITRECD as generation options. Refer to the VisualAge Generator online help system for additional information on how to initialize records using a SET record EMPTY. Use the phrase SET record EMPTY for scanning the helps.

The abend can also occur when SET record EMPTY is used or when /INITADDWS and /INITRECD are used if one of the following is true:

- There are redefined records with different data types or variable field boundary alignments from the original record
- The primary working storage record receives a transferred record that contains different data types or variable-field boundary alignments from the original record.

See “Languages Available” on page 587 for a list of valid NLS language codes.

LE Runtime Messages

Only the most frequently occurring LE runtime messages are listed in this section. If you receive other runtime messages that start with IGZ or if you need a more complete explanation of one of the messages, refer to the debugging manual for your release of LE.

IGZ0033S **An attempt was made to pass a parameter address above 16 megabytes to AMODE(24) program program-name.**

Explanation: An attempt was made to pass a parameter located above the 16-megabyte storage line to a program in AMODE(24). The called program will not be able to address the parameter.

Programmer Response: If the calling program is compiled with the RENT option, the DATA(24) option may be used in the calling program to make sure that its data is located in storage accessible to an AMODE(24) program. If the calling program is compiled with the NORENT option, the RMODE(24) option may be used in the calling program to make sure that its data is located in storage accessible to an AMODE(24) program. Verify that no linkedit, binder or genmod overrides are responsible for this error.

System Action: The application was terminated

IGZ0064S **A recursive call to active program program-name in compilation unit compilation-unit was attempted.**

Explanation: COBOL does not allow reinvocation of an internal program which has begun execution, but has not yet terminated. For

example, if internal programs A and B are siblings of a containing program, and A calls B and B calls A, this message will be issued.

Programmer Response: Examine your program to eliminate calls to active internal programs.

System Action: The application was terminated.

IGZ0066S **The length of external data record data-record in program program-name did not match the existing length of the record.**

Explanation: While processing External data records during program initialization, it was determined that an External data record was previously defined in another program in the run unit, and the length of the record as specified in the current program was not the same as the previously defined length.

Programmer Response: Examine the current file and ensure the External data records are specified correctly.

System Action: The application was terminated.

IGZ0075S Inconsistencies were found in EXTERNAL file file-name in program program-name. The following file attributes did not match those of the established external file: attribute-1 attribute-2 attribute-3 attribute-4 attribute-5 attribute-6 attribute-7

Programmer Response: Correct the external file. For a summary of file attributes which must match between definitions of the same external file, see *IBM COBOL Language Reference*.

System Action: The application was terminated.

Explanation: One or more attributes of an external file did not match between two programs that defined it.

Common COBOL Abend Codes

Only the most frequently occurring abend codes are listed in this section. If you receive another abend code or if you need a more complete explanation of one of the messages, refer to the debugging manual for your release of COBOL.

User 1015

This COBOL abend code indicates there has been a recursive call. For example, A calls B and B in turn calls A. COBOL does not support this.

Common VSE CICS and VSE Batch Runtime Messages

Only the most frequently occurring VSE CICS and VSE batch runtime messages are listed in this section. If you receive another VSE CICS or VSE batch runtime message that starts with DFH or if you need a more complete explanation of one of the messages, refer to the VSE CICS and VSE batch messages and codes manual for your release of VSE CICS or VSE.

DFH2016 *time applid* TRANSACTION *tranid*
CANNOT RUN BECAUSE
PROGRAM *prognam* IS NOT
AVAILABLE

Explanation: Transaction *tranid* is not executable because the initial program for transaction *tranid* is not available. Possible reasons for this are:

- The program is missing.
- The installed program definition is missing.
- The program is disabled.
- The program name in the installed transaction definition is invalid.

System Action: Other processing continues.

User Response: Depending upon the cause of the error, the action is as follows:

- Load the program into the CICS/VSE program library.
- Create an installed program definition for the program.
- Enable the program.
- Use a valid program name in the installed transaction definition.

Have your system administrator check the PPT entries. Be sure the program is in a library in the DFHRPL concatenation.

Destination: Terminal end user

Module: DFHACP

DFH2206 *time applid* TRANSACTION *tranid*
HAS FAILED WITH ABEND
abcode. RESOURCE BACKOUT
WAS SUCCESSFUL. *condmsg*

Explanation: Transaction *tranid* ends abnormally with abend code *abcode*. All recoverable resources have been successfully backed out following the abend.

abcode is either a CICS/VSE transaction abend code (see "Transaction Abend Codes" in *VSE/ESA Messages and Codes, Volume 1* or a user abend code generated by a CICS/VSE ABEND ABCODE(*abcode*) command. This command is issued either by a user program or by an IBM program (for example, a programming language library module).

If possible, a conditional message *condmsg* from the system that is ending is appended to this message.

System Action: Message DFH2236 is sent to the master terminal operator (destination CSMT). Normal abend processing continues.

User Response: Use the abend code, *abcode*, to diagnose the problem. If the abend is issued by an IBM program product other than CICS/VSE, the code is documented in the library of that other product.

Resubmit the transaction after the cause of the original abend has been removed.

Destination: Terminal end user

Module: DFHACP

DFH2236 *time applid* TRANSACTION *tranid*
ABEND *abcode* IN PROGRAM
progrname TERM *termid* BACKOUT
SUCCESSFUL
{BATCHID=*batchid*|SYSTEM*sysid*
SENT MESSAGE SENSE CODE
(*cccccc*):*message*}

Explanation: Transaction *tranid* ends abnormally with abend code *abcode* in program

progrname. The recoverable resources have been successfully backed out following the abend.

BATCHID=*batchid* is added when *tranid* is a shared database mirror transaction.

SYSTEM *sysid* is added when any of the following is true:

- *tranid* is started by transaction routing from another CICS/VSE system on an MRO link.
- *tranid* is started by transaction routing from another CICS/VSE system on an ISC APPC link.
- *tranid* is the mirror transaction in an MRO session using function shipping.
- *tranid* is a transaction involved in distributed transaction processing.

In the case of shared databases, a mirror transaction ends abnormally. The message describes what has happened to any DL/I updates performed by batch program *batchid* (following the specified checkpoint, if any).

Terminal *termid* represents the connection between the batch partition and CICS/VSE rather than a real terminal.

The *batchid* value provided is x followed by t, where x is the jobname.stepname.procname. This is true unless a CHKP call has been issued by the batch program, in which case x is the checkpoint identifier.

t is the time (hh:mm:ss) at the start of the job or the latest checkpoint.

In the case of an MRO or an ISC APPC (parallel sessions), transaction *tranid* ends abnormally with abend *abcode* in program *progrname*.

termid is a terminal identifier (transaction routing) or a session identifier. *sysid* is the identifier of the linked CICS/VSE system, and *cccccc* represents the SNA sense bytes.

The display ends with the message issued by the linked CICS/VSE system.

System Action: If possible, message DFH2206 is sent to the terminal end user. Normal abend processing continues.

User Response: For an explanation of the CICS/VSE abend code *abcode*, see "Transaction Abend Codes" in *VSE/ESA Messages and Codes, Volume 1*. If the code is not documented, it could be a COBOLII abend or a user code.

If *abcode* is a COBOLII abend, diagnostic information is written into the temporary storage queue *CEBRname*, where *name* is the name of the terminal. Use this information in your investigation of the problem.

If *abcode* is a user code, consult the programmer responsible for this area.

For an explanation of the SNA sense bytes *ccccccc*, see the *Systems Network Architecture Reference Summary* manual. Ignore the last two bytes.

Destination: MT

Module: DFHACP

Common VSE CICS and VSE Batch Abend Codes

Only the most frequently occurring VSE CICS and VSE batch runtime messages are listed in this section. If you receive another VSE CICS or VSE batch abend or if you need a more complete explanation of one of the abend codes, refer to the VSE CICS and VSE batch messages and codes manual for your release of VSE.

Depending on your diagnostic options, information might be available on an error destination queue or in a VSE journal. See "Controlling Error Reporting in VSE CICS" on page 405 and the appropriate ELA messages in "Messages" on page 590.

ACN1 The DFHCNV conversion table cannot be loaded. If the program running under CICS OS/2 attempts a DPL to a generated COBOL program running under VSE CICS, you need to provide the CICS conversion table, DFHCNV. Although the table is not used by VisualAge Generator Server, it is required by CICS/VSE.

AEY9 An incorrect EXEC CICS command was issued. This abend code can occur if SQL/DS is not running.

If you attempt to use the VSE CICS report control facility and do not have it installed on your CICS system, you might receive a CICS abend code AEY9.

APCT A requested module cannot be located in the PPT or in the program library.

ASRA A program check occurred. Some of the reasons this can occur for a VisualAge Generator program are shown below:

- Incorrectly linked CSP/370RS modules. If register 1 contains ELA1, see the information for ELA1 in "VSE CICS and VSE Batch Environments" on page 451.

- Data not initialized or data initialized to incorrect values. If the error occurred as a result of a data exception, see the explanation for "System 0C7" in "Common System Abend Codes for All Environments" on page 457.

ATDD The program attempted to process a transient data queue that is disabled. This can occur for a program file associated with a transient data queue or for the transient data queue used for error diagnostic information.

AXFQ The most common cause is the result of INBFMH not being specified to equal to ALL in the profile associated with the CICS mirror transaction (CPMI).

If the program running under CICS OS/2 attempts a DPL to a generated COBOL program running under VSE CICS, you need to ensure that the profile associated with CPMI transaction is defined with INBFMH=ALL.

COBOL Abends under CICS

- 1009** A program has a dynamic storage requirement greater than 64KB, but was compiled with the DATA(24) compiler option. Compile the module again with the DATA(31) compiler option.
- 1029** Either a PPT entry for a program attached through a COBOL dynamic call is not found or the module being invoked cannot be found in the VSE CICS or VSE batch partition program library search string. Additional information can be retrieved by entering transaction CEBR on the terminal where the error occurred.
- 1033** A 31-bit parameter was passed to a 24-bit program. This abend can occur if VisualAge Generator Server is operating in 31-bit mode and you are running a 24-bit program that uses a VisualAge Generator table program link edited with AMODE(24). The table program must be linked with AMODE(31) to receive the data passed to it by ELAASTBC (the VisualAge Generator Server table program). DATA(24) and RMODE(24) should be used if the program should reside below the 16MB line.

Common Runtime Messages for DL/I DOS/VS

Only the most frequently occurring DL/I DOS/VS runtime messages are listed in this section. If you receive other runtime messages that start with DLZ or if you need a more complete explanation of one of the messages, refer to the DL/I DOS/VS messages and codes manual for your release of DL/I DOS/VS.

DLZ109I **DL/I Parameters above 16MB, task terminated**

message occurs if you are running DL/I DOS/VS and your DL/I program was not generated/compiled with the DATA(24) option.

Explanation: DL/I has detected that parameters are located above the 16MB line of storage. This

Common Runtime Messages for SQL/DS

Only the most frequently occurring SQL/DS runtime messages are listed in this section. If you receive other runtime messages that start with ARI or if you need a more complete explanation of one of the messages, refer to the SQL/DS messages and codes manual for your release of SQL/DS.

ARI0415D **Database Server *servername* is not ready. Enter WAIT or CANCEL.**

generated program issued an SQL request.

Explanation: The SQL/DS system was not started in multiple user mode before the

Part 10. Administering on VM Systems

Chapter 47. General System	
Considerations for VM Systems	467
Virtual Machine Size	467
Considerations that Affect Performance	467
Generation and Compiler Options	467
Files and Databases	468
Defining and Managing VSAM Program	
Data Files	468
Defining Data Files	469
Loading Data into a VSAM File	470
Support for DBCS Terminals	471
Extended Addressing Considerations for	
VisualAge Generator Server	471
Database Considerations	472
Preparing Programs	472
Checking Access Authorization	473
Moving Program Packages Among VM	
Systems	473
Fold Tables	474
Backing Up Data	474
Customizing VisualAge Generator Server	474
Chapter 48. System Considerations for	
VM CMS and VM batch	475
Required and Optional File Descriptions	475
Printing Considerations for VM CMS and	
VM batch	476
Making VisualAge Generator Server,	
COBOL, and the Generated Modules	
Available	477
Recovery Considerations	477
Using VSAM Program Data Files in VM	478
Considerations for Using DB2/VSE in VM	479
Connection Considerations for SQL	
Programs	479
Starting SQL Programs in VM	481
Recovery and Database Integrity	
Considerations	482

Chapter 47. General System Considerations for VM Systems

This chapter describes the system requirements and considerations for administering VisualAge Generator Server in all of the supported VM environments.

The following information is discussed:

- Virtual Machine Size
- Considerations that affect performance
- Defining and loading VSAM program data files
- Support for DBCS terminals
- Extended addressing considerations for VisualAge Generator Server
- SQL considerations
- Fold tables
- Backing up data
- Customizing VisualAge Generator Server

Virtual Machine Size

The storage needed by the program added to the storage needed for VisualAge Generator Server for MVS, VSE, and VM determines the virtual machine size needed to run a VisualAge Generator Server for MVS, VSE, and VM program. Because VisualAge Generator Server for MVS, VSE, and VM dynamically loads some of the modules it uses, it is difficult to determine exactly how much storage is needed for a particular program. The minimum needed for any program is 4MB (megabytes). However, more storage is needed to run the COBOL compiler in the preparation environment.

Considerations that Affect Performance

Specifying certain generation and compiler options and making reentrant programs resident in memory can affect the performance of VisualAge Generator programs.

Generation and Compiler Options

Specifying the following generation options might improve runtime performance:

- /INEDIT=INONLY
- /MATH=COBOL
- /NODEBUGTRACE
- /NOINITADDWS
- /NOINITRECD
- /NOLEFTJUST
- /NONULLFILL
- /NONUMOVFL
- /NOSETFULL
- /NOVALIDMIX

Specifying the following compiler options might improve runtime performance:

- NOFDUMP
- NOSSRANGE
- NOTEST
- OPT

Note: Refer to the *VisualAge Generator Generation Guide* for details on these options.

Files and Databases

Standard tuning techniques (such as buffering) can be used with files and databases used by generated COBOL programs.

Defining and Managing VSAM Program Data Files

This section describes the procedure for defining VSAM data files for use in a VisualAge Generator program.

You use files when testing or running a program. When you define a VisualAge Generator program, consider the organization of the file.

A file can be organized as follows:

- Serial (VSAM ESDS or CMS files)
- Relative record (VSAM RRDS)
- Indexed (VSAM KSDS)

A serial file contains records that are arranged sequentially and retrieved in the same order as they are stored.

A relative record file contains records that are located by unique data items called relative record numbers. A relative record number contains a numeric value that is used to locate a specific record in a file, but the relative record number does not have to be a part of the record.

An indexed file contains two parts: the index (key) part and the data part. Direct access of records is available with an indexed file. The index or key value is used to locate a specific record in a file. The index must be contained in the record.

Note: Before AMSERVs can successfully execute, you must enter DLBLs for the master and user VSAM catalogs.

Defining Data Files

Before VSAM files can be used, they must be defined. Figure 81 shows the contents of the VM CMS file DEFTEST AMSERV that defines a VSAM indexed file, a relative record file, and a serial file.

Where:

```
/* FOLLOWING IS A KEY-SEQUENCED DATA SET */
DEFINE CLUSTER (NAME(VM1.USER.KSDS) -
               RECORDS(100 100) -
               VOL(xxxxxx) KEY(7 0) -
               RECORDSIZE(yyy zzz) -
               INDEXED) -
          DATA (NAME(VM1.USER.KSDS.DATA)) -
          INDEX (NAME(VM1.USER.KSDS.INDEX))
/* FOLLOWING IS A RELATIVE RECORD DATA SET */
DEFINE CLUSTER (NAME(VM1.USER.RRDS) -
               RECORDS(100 100) -
               VOL(xxxxxx) -
               RECORDSIZE(yyy zzz) -
               NUMBERED) -
          DATA (NAME(VM1.USER.RRDS.DATA))
/* FOLLOWING IS A ENTRY-SEQUENCED DATA SET */
DEFINE CLUSTER (NAME(VM1.USER.ESDS) -
               RECORDS(100 100) -
               VOL(xxxxxx) -
               RECORDSIZE(yyy zzz) -
               NONINDEXED) -
          DATA (NAME(VM1.USER.ESDS.DATA))
```

Figure 81. DEFTEST AMSERV

xxxxxx is the volume identifier

yyy is the average record size

zzz is the maximum record size

You can define the preceding sample test files by entering the following VM CMS command:

```
AMSERV DEFTEST
```

This command defines the VSAM files and produces the VM CMS file DEFTEST LISTING. Check this file for errors.

See *VSE/VSAM Commands and Macros* for more information about defining a data set.

In order for a defined VSAM file to be used, a DLBL must be issued for it. For example:

```
DLBL ddname x DSN CSP1 USER KSDS (VSAM
```

Where:

DDNAME is the name that you use to refer to the file

x is the mode of the accessed disk containing the file

See the *VM/ESA CMS Command Reference* for more information about the DLBL command.

Loading Data into a VSAM File

If you are using a VSAM indexed file (KSDS) and want to open it for input, then you must initialize the file with at least one record. This is due to a VSAM restriction that prevents a file from being opened for input if it is empty. Once the file contains data, then it can be opened for input, output, or both.

After a file has been defined, you can use the AMSERV REPRO function to load data into the file. The following is an example of the VM CMS file named REPROI AMSERV:

```
REPRO INFILE (DATA ENV (BLKSZ(256) -  
RECFM (F))) -  
OUTFILE (TESTKEY)
```

You then need to enter the following VM CMS DLBL command for the input file DATA and the output file TESTKEY:

```
DLBL DATA A CMS DATA FILE  
DLBL TESTKEY x DSN CSP1 USER KSDS (VSAM
```

Where: *x* is the mode of the accessed disk that contains the file TESTKEY.

Edit the data file and place at least one record (dummy or valid) in it.

Enter the following VM CMS command:

```
AMSERV REPROI
```

The preceding commands load the VM CMS file DATA FILE A into the VSAM key-sequenced file TESTKEY and produce the VM CMS file REPROI LISTING. Check this file for errors.

Refer to the *VSE/VSAM Commands and Macros* manual for more information about how to REPRO a data set.

If you are using a VSAM relative record file, you can open the file for input or output even if it is empty. However, if the file is empty, it must be closed after records are added before those records are available for input.

To add records to a relative record file that is empty, you can write a VisualAge Generator program that uses the ADD process option to add a record, followed by a CLOSE process option to close the file. Alternatively, you can use File Maintenance in VisualAge Generator to add the records.

Note: If the file is to be indexed (key-sequenced), the VM CMS file must be sorted on its key field before it is loaded into the VSAM file with the AMSERV REPRO function.

Support for DBCS Terminals

VisualAge Generator Server provides support for the IBM Personal System/55 and the IBM 5550 family of terminals (emulating an IBM 3270 device). In addition to the basic hardware, this support uses character set F8 and four hardware attributes for double-byte character set (DBCS). The extended attributes are shift-out (SO) and shift-in (SI) enable, field outlining, color, and extended highlighting.

For the MVS CICS, MVS/TSO, VSE, and VM CMS environments, VisualAge Generator Server sends hardware attributes to the terminal only if the terminal supports them. The attributes are ignored if the terminal does not support them.

Extended Addressing Considerations for VisualAge Generator Server

Some of the code provided by VisualAge Generator Server can run in extended addressing mode. This section describes considerations for using the extended addressing mode.

Most of the code shipped with VisualAge Generator Server runs in 31-bit addressing mode and resides above the 16MB line.

Most of the storage acquired by VisualAge Generator Server is above the 16MB line unless the first VisualAge Generator-generated program in the run unit is link-edited with AMODE(24) or is compiled with the /DATA=24 generation option. The AMODE(24) program attribute specifies that the program runs in 24-bit addressing mode. Refer to the *VisualAge Generator Generation Guide* for more information on AMODE(24) or the /DATA generation option.

Database Considerations

This section discusses preparing programs and checking access authorization to database resources when using the following:

- DB2 on MVS systems
- DB2/VSE on VSE systems and moving program packages among VSE systems
- SQL on VM systems

Preparing Programs

Before running a program, the SQL* statements need to be analyzed and prepared. You can choose to have your programs prepared by DB2 (the default on MVS systems), DB2/VSE (the default on VSE and VM systems), or an ANSI SQL database manager.

If you specify the /ANSISQL generation option, any SQL statements in the generated program are generated using the ANSI SQL format. If you use ANSI SQL, you need to tailor the templates, procedures, and execs for your database manager.

If you use DB2, you also need to bind the DB2 program plan. Refer to the a document for information on binding DB2 program plans.

If your programs run in the MVS/TSO, MVS batch, or IMS BMP environments, you might also need to tailor the runtime JCL and CLIST templates. If your programs run in the VM CMS or VM batch environment, you might also need to tailor the runtime exec templates. Refer to the *VisualAge Generator Generation Guide* for additional information on tailoring runtime JCL, CLIST, and exec templates.

Checking Access Authorization

The database manager checks whether program users have the authority to access tables or run programs. The type of checking done varies depending on your system and the processing mode.

To run a DB2/VSE program, program users must be granted authority to run the program and to process any tables accessed using dynamic SQL statements in the program.

Moving Program Packages Among VM Systems

If you have generated an SQL program and run the preparation job or REXX exec created by VisualAge Generator Developer, your program and the SQL package that is created during the DB2/VSE preprocessing step contain consistency tokens based on a time stamp. The program and its associated SQL package must have consistency tokens that match.

On VSE systems, you must ensure that your program and its SQL package are created in the same preparation job stream.

On VM systems, you must ensure that the program module and the SQL package you are using were created by the same invocation of the preparation REXX exec. Refer to the CTOKEN parameter description of the SQLPREP command in the *DB2/VSE Application Programming for IBM VM Systems* manual for more information.

If you need to run your program on a different VM system than the one for which it was generated, do the following:

1. Use the CMS SENDFILE command to send a copy of the LOADLIB to the VM system where you plan to run your program. The CMS LOADLIB command can be used to copy individual members of a LOADLIB to a different LOADLIB. This is useful to make a smaller production LOADLIB or to combine members from multiple LOADLIBs into one LOADLIB.

Note: This step is optional because it is possible to use the same load library for different VM systems.

2. Use the DB2/VSE DBS utility, UNLOAD PACKAGE, to copy the SQL package to a sequential file. Then use the RELOAD PACKAGE utility to read the package back into the database server on the VM system where you intend to run your program. Refer to the *DB2/VSE Database Services Utility for IBM VM Systems* manual for more information on unloading and reloading packages.

Fold Tables

VisualAge Generator Server does folding (uppercase translation) using a 256-byte translate table. VisualAge Generator Server uses the translate table at offset X'0C' in CSECT ELANCccc where ccc is the language code.

Backing Up Data

You should regularly back up your data. This includes all files related to VisualAge Generator Server, private libraries, user-created data files, and user load libraries. System services are provided to back up and restore user libraries.

Customizing VisualAge Generator Server

Customizing VisualAge Generator Server consists of performing some of the same procedures used to install the product on the system. These procedures are described in the *Program Directory for VisualAge Generator Server* for your MVS, VSE or VM system. The program directory contains information on changing system options.

Chapter 48. System Considerations for VM CMS and VM batch

This chapter provides additional administrative information that applies to the VM CMS and VM batch environments.

The following information is discussed:

- Required and optional file descriptions
- Printing considerations
- Making VisualAge Generator Server, COBOL, and the generated modules available
- Recovery considerations
- Using VSAM program data files
- Considerations for using SQL

Required and Optional File Descriptions

VisualAge Generator Server requires the following files:

File Name	Description
EZEPRINT	<p>This file is used when printing from a program that displays printer maps. The sample runtime REXX exec is set up to print to the virtual printer; however, you can customize the exec to print to another destination. EZEPRINT can be allocated to either a file or to a virtual printer. The file must have a VBA (variable-blocked ANSI) record format.</p> <p>The maximum record length that a generated program can write to the print data set is 654 bytes for DBCS maps and 137 bytes for SBCS maps. The record length includes 4 bytes for the variable length record header, 1 byte for the American National Standards printer-control character, and the print line for the print map. The DBCS record length is longer than the printer line length because the print line can contain outlining control characters and shift-out (SO) and shift-in (SI) characters that are not displayed on the device. The logical record length defined for the data set must be greater than or equal to the length of the longest line written by the program, including the DBCS SO/SI characters.</p> <p>If you are using VisualAge Generator Server to print to a file destination other than EZEPRINT, the characteristics of that</p>

file should be the same as EZEPRINT. The file destination can be specified either during resource association at generation time, or at run time with EZEDESTP special function word.

- ELASNAP** This is the file where VisualAge Generator Server for MVS, VSE, and VM will direct a snap dump if a VisualAge Generator Server for MVS, VSE, and VM abend condition occurs. ELASNAP is normally allocated to a virtual printer. The file must have a VBA (variable-blocked ANSI) record format. ELASNAP can be allocated to suppress snap dumps using the DUMMY operand of the CMS FILEDEF command.
- ELAPRINT** This system output file is used by generated programs. Specify ELAPRINT with RECFM=FBA.

The following files are optional for VisualAge Generator Server for MVS, VSE, and VM:

File Name	Description
ELATRACE	This file is the trace control file for the VM CMS environment. The attributes for this data set are LRECL=80 and RECFM=FB. The trace filters are specified in the ELATRACE data set.
ELATOUT	The output of the VisualAge Generator Server trace facility is sent to this data set in the VM CMS and VM batch environments. The attributes for this data set are LRECL=133 and RECFM=F.

Printing Considerations for VM CMS and VM batch

Generated program printing is routed to the file that is specified as the resource associated to EZEPRINT. To print directly to the CMS virtual printer, add the following statement to your runtime exec:

```
'FILEDEF EZEPRINT PRINTER (RECFM VBA LRECL 137'
```

To print to a disk file, add the following statement to your runtime exec:

```
'FILEDEF EZEPRINT DISK fn ft fm (RECFM VBA LRECL 137'
```

If a file destination other than EZEPRINT is specified either during generation or at run time (using the EZEDESTP special function word), output is directed to the specified destination. For example, if the program developer generated the program with EZEDESTP specified as ALTPRINT, you must allocate the file name ALTPRINT and direct the print output to it before running the program. If you do not specify the proper file name, the print connection is not successful. Refer to the online help system for more information on the EZEPRINT and EZEDESTP special function words.

Terminal printing is not directly supported in VM CMS.

Making VisualAge Generator Server, COBOL, and the Generated Modules Available

For users to run your program, they must have access to the runtime exec for your program, the program LOADLIB, and any data files that your program uses. Either copy these resources to production minidisks or SFS directories that the users normally access, or provide an access exec that links and accesses the resources containing these files and invokes your program runtime exec.

Also, during VisualAge Generator Server for MVS, VSE, and VM installation, the ELASETUP EXEC and ELACLEAN EXEC should have been copied to a minidisk or an SFS directory that users normally access. These execs access and release the VisualAge Generator Server for MVS, VSE, and VM production disk that is required for preparation and execution of VisualAge Generator programs.

Recovery Considerations

A logical unit of work (LUW) for a VM CMS or VM batch program ends whenever a commit point or a rollback occurs.

A commit point occurs in VM CMS and VM batch when one of the following occurs:

- The top-level program in a run unit ends successfully.
For VM CMS or VM batch, a run unit consists of all VisualAge Generator programs and non-VisualAge Generator programs that transfer control among themselves using an XFER, DXFR, or CALL statement. For non-VisualAge Generator programs, this also includes any transfer that uses an OS XCTL macro.
- A transfer using an XFER statement occurs for a segmented or single segment program.
- A program transfer using an XFER statement and the /SYNCXFER generation option is specified for the transferred-from program.
- A program calls the EZECOMIT service routine.
For VM CMS and VM batch, VisualAge Generator programs issue a commit point only if the program has made changes to an SQL table. A commit point does not occur for changes to an SQL table made by a non-VisualAge Generator program.

A commit point occurs in VM CMS only when one of the following occurs:

- A program uses a CONVERSE process option, and any of the following is set to 1:
 - EZESEGM special function word (segmented mode)
The EZESEGM special function word defaults to 1 if the program is defined as segmented.
 - EZECNVCM special function word (CONVERSE commit)

The best time for a commit point to occur is after terminal output and before the next terminal input. A commit point at terminal I/O synchronizes updates to the database and confirmation messages to the program user.

A rollback in VM CMS and VM batch occurs because of the following:

- A program calls the EZEROLLB service routine.
For VM CMS and VM batch, VisualAge Generator programs issue a rollback only if the program has made changes to an SQL table. However, a rollback does not occur if changes were made to an SQL table by a non-VisualAge Generator program.
- A program ends because of an error condition.

When a rollback occurs, all changes that were made to databases since the start of the LUW are backed out.

Using VSAM Program Data Files in VM

VSAM program data files must be defined before your VM CMS or VM batch program can use them. Use the AMSERV program to define a user VSAM data file to VM CMS. See “Defining and Managing VSAM Program Data Files” on page 468 for information on defining VSAM data sets, defining alternate indexes, and for information on loading VSAM data sets.

You can either allocate user data sets by using the VM CMS DLBL command or let VisualAge Generator Server allocate them dynamically. Refer to the *VisualAge Generator Design Guide* for more information on file allocation.

If you are using COBOL, you must specify /FILETYPE=VSAMRS on the associate statement in the Resource Association file for all VSAM files. COBOL does not support VSAM files when the virtual machine is running in XA, ESA or XC mode. Because VisualAge Generator programs cannot be run in a 370 mode virtual machine, /FILETYPE=VSAMRS must be specified so that VisualAge Generator Server for MVS, VSE, and VM performs the VSAM file I/O.

Considerations for Using DB2/VSE in VM

This section discusses the following considerations for programs using DB2/VSE in VM CMS and VM batch:

- Connection considerations for SQL programs
- Starting SQL programs in VM CMS and VM batch
- Recovery and database integrity considerations

For information on designing and generating DB2/VSE programs for the VM environment, refer to the *VisualAge Generator Design Guide* and the *VisualAge Generator Generation Guide*.

For information on preparing DB2/VSE programs for running in the VM environment, refer to “Preparing to Run an SQL Program” on page 493.

Connection Considerations for SQL Programs

In VM CMS and VM batch, the DB2/VSE CONNECT statement is not required to establish a connection between DB2/VSE and your program. A generated DB2/VSE program can establish the database connection by one of the following methods:

- VM support for DB2/VSE issues an implicit CONNECT to establish the database connection at the first SQL statement. DB2/VSE accepts the password verification of the VM virtual machine and uses the VM user ID as the DB2/VSE user ID.
- Calling the EZECONCT service routine to connect the specific user ID to a database server. This call must be issued before any other SQL request in the run unit.
- By specifying the user ID, password, and the database name during generation using the symbolic parameters SQLUSRPW and SQLDBNAM. If SQLUSRPW, SQLDBNAM, or both are specified during generation, they are used as the default user ID, password, and database name when the program is executed. If the user ID, password, or database name used during generation are different than the user ID, password, and database name used during execution of the program, then the program runtime REXX exec must be modified to either change or remove the generation defaults.

In Figure 82 on page 480, the program was generated using an SQLUSRPW value of MUID/MYPWD and an SQLDBNAM value of SQL_VM. To modify the user ID and password for execution, the EZE_SQL_SID variable must be changed to the new user ID and password or set to a value of null (''). To modify the database name for execution, the EZE_SQL_SDB variable must be changed to the new database name or set to a value of null (''). If a

null value is specified for either EZE_SQL_SID or EZE_SQL_SDB, one of the other methods is used to establish the SQL database connection.

```

/*REXX*/
/*
=====
      EFK2CERB - EXECUTE VM APPLICATION

              MAIN APPLICATION: A2CONCT
      TARGET ENVIRONMENT: VMCMS
              GENERATION DATE: 08/05/96
              GENERATION TIME: 07:27:25
              SQL USED: Y
=====
*/
ADDRESS COMMAND
PARSE ARG INVARS '$$LA$V$A$R$$:' ELAVARS
PARSE VAR ELAVARS ELA_APPL_FUNCTION ',' ELA_APPS ',' ,
              ELA_APPL_VMLoadLIB ',' ,
              ELA_SQL_USED ',' ELA_SQL_SID ',' ELA_SQL_SDB
PARSE SOURCE . . ELA_EXEC_FN .
PARSE UPPER VAR INVARS . 'SID=' IN_SID .
PARSE UPPER VAR INVARS . 'SDB=' IN_SDB .
EZE_APPL_VMLoadLIB = 'EZEAPPL'
EZE_SQL = 'Y'
IF IN_SID = '' THEN
  EZE_SQL_SID = 'MYUID/MYPWD'
ELSE
  EZE_SQL_SID = IN_SID
IF IN_SDB = '' THEN
  EZE_SQL_SDB = 'SQL_VM'
ELSE
  EZE_SQL_SDB = IN_SDB

```

Figure 82. Modifying the Runtime REXX Exec for SQL Database Connection

- By specifying the user ID, password, and the optional database name on the invocation of the runtime REXX exec for the generated program. VisualAge Generator Server uses the values you specify in the SID= and SDB= parameters of the generated program runtime REXX exec to connect the user to a database during execution of the program. The SID= and SDB= parameters are optional. If you do not specify either parameter, your program accesses the database specified in the last DB2/VSE SQLINIT EXEC processed on your virtual machine. The database is accessed in multiple user mode using your VM user ID as the DB2/VSE connection ID. The format of the SID= and SDB= parameters is shown below:

SID=uuuuuuu/pppppppp

SID stands for the DB2/VSE connection ID, *uuuuuuuu* is a 1- to 8-character SQL user ID, and *pppppppp* is a 1- to 8-character SQL user password. If

SID= is specified, VisualAge Generator Server is called to issue a CONNECT SQL statement with the specified user ID and password prior to accessing the SQL database during the execution of the generated program. If you do not specify SID=, no CONNECT is done and DB2/VSE uses your VM user ID and password for the DB2/VSE user ID and password.

```
SDB=sssssss
```

SDB stands for the SQL/DS database name and sssssss is a 1- to 8-character database name. The default name is SQLDBA if you are in single user mode. If you do not specify SDB= in multiple user mode, the default database is the database specified on the last DB2/VSE SQLINIT EXEC processed on the virtual machine. If you specify SDB= in multiple user mode, a SQLINIT EXEC is issued to identify the database to be used for the generated program.

The following example shows how to use the SID= and SDB= parameters to specify a user ID and password to connect to the SQL_VM database during the execution of a generated program called MYAPP.

```
MYAPP SID=MYUID/MYPWD SDB=SQL_VM
```

A generated DB2/VSE program running under VM CMS or VM batch can access multiple DB2/VSE program servers. A generated DB2/VSE program can switch to a different database by calling the EZECONCT service routine. However, you must ensure that the program user has database CONNECT authority for this function.

Starting SQL Programs in VM

The way you start a generated DB2/VSE program in VM CMS or VM batch depends on whether you run in multiple or single user mode.

Use multiple user mode if the relational database is to be shared among users, programs, or both. For multiple user mode, use the same runtime REXX exec that you would for an ordinary VM CMS or VM batch program. The system operator must have started DB2/VSE before your job runs in multiple user mode.

Use single user mode if no other programs or users need to obtain access to the relational database while your program runs. Use the same runtime REXX exec that you would use for an ordinary VM CMS or VM batch program. The runtime REXX exec will issue an SQLSTART to initialize the DB2/VSE database. DB2/VSE then passes control to VisualAge Generator Server for MVS, VSE, and VM to start your program.

Recovery and Database Integrity Considerations

VisualAge Generator programs can use all the recovery and data integrity features provided by DB2/VSE.

Relational databases are recoverable resources. If your program makes changes to a relational database, the changes are not committed to the database until the end of a logical unit of work (LUW). If your program ends abnormally before the end of an LUW, all changes that were made since the beginning of the LUW are backed out. See “Recovery Considerations” on page 477 for information describing when an LUW ends.

Part 11. Preparing and Running Generated Applications on VM Systems

Chapter 49. Output of Program Generation on VM Systems	485
List of Program Preparation Steps after Program Generation.	485
Outputs of Generation	485
Objects Generated for Programs.	487
Application COBOL Program	487
Sample Runtime REXX exec	488
Objects Generated for Tables	488
Table COBOL Program	488
Objects Generated for Map Groups.	488
Online Print Services Program	488
Batch Print Services Program.	488
Map Group Format Module	488
Objects Generated for All Part Types	489
Listing File.	489
Parts Control File	489
Preparation Command File	489
Preparation REXX exec.	489
Chapter 50. Preparing Generated Programs on VM Systems	491
Running the Preparation REXX Exec	491
Analyzing the Preparation Listing	491
Analyzing the SQL Precompile Step	491
Analyzing the Compile Step	491
Analyzing the Link-Edit Step.	492
Link-Edit Errors Related to Missing Modules	492
Other Link-Edit Errors	493
Preparing to Run an SQL Program	493
Granting Authority to Program Users	493
Defining Synonyms for SQL Tables.	493
Using LE Runtime Options for VM CMS and VM batch.	494
Chapter 51. Modifying the Sample Runtime REXX Execs	497
Tailoring Sample REXX Exec Templates	
Before Generation	498
Modifying the Sample Runtime REXX Execs	
After Generation	500
File Allocation Considerations	500
General File Allocation Considerations	500
Files Used By The Main Program	500
Using EZEDEST or EZEDESTP for File Allocation.	501
Files Used by Other Known Programs	501
Files Used By EZEAPP Programs	501
Required and Optional File Allocations.	502
SQL Considerations	502
Runtime REXX EXEC Examples.	502
Chapter 52. Running a Generated Program on VM Systems	511
Making VisualAge Generator Server for MVS, VSE, and VM, COBOL, and the Generated Modules Available	511
Running Main Programs under VM	512
VM Saved Shared Segments	512
Chapter 53. Moving Prepared Programs to Other Systems from VM Systems	515
Moving Prepared Programs To Another VM System	515
Maintaining Backup Copies of Production Libraries	515
Chapter 54. Execs for Preparing and Running Programs on VM Systems	517
ELACALLC EXEC	518
ELAPREP EXEC	521
ELARUN EXEC	527
ELASQLAA EXEC	528

Chapter 49. Output of Program Generation on VM Systems

This chapter provides an overview of the output files from program generation. The output files are transferred to the VM host system and stored on minidisks and SFS (shared file system) directories. Use these files to prepare the programs for production in the same way you would prepare COBOL programs you might write directly. Preparation steps include running precompilers and compilers, and performing link-edits for the target runtime environment.

List of Program Preparation Steps after Program Generation

VisualAge Generator Server supports program preparation and installation in the VM environments using the generated preparation REXX exec and the product execs shipped with VisualAge Generator Server. You must perform the steps listed in Table 64 before you can run your program in a VM target environment.

Table 64. Preparation Steps for VM Environments

Preparation Step	Environment
Transfer from workstation to the host	All
SQL precompile	SQL programs only
COBOL compile	All
Link	All

For information about generating programs and generation options, refer to the *VisualAge Generator Generation Guide*.

Outputs of Generation

After you have generated a program, a number of objects must be transferred to the host system as files on a minidisk or a SFS directory. On the VM system, these files need to be prepared before the program can be run.

The following tables provide information about the types of files produced by generation, including:

- Type of object produced
- OS/2 file name and extension
- Runtime environments for which the object is produced

- Generation option that controlled whether the object is produced
- Whether the object can be modified after generation is performed

Refer to the *VisualAge Generator Generation Guide* for more information on controlling and modifying generation and preparation of VM objects.

“Objects Generated for Programs” on page 487 provides a description of each object.

Table 65. Objects Produced for Programs by the Generation Facility for VM

Object Type	OS/2 File Name and Extension	VM File Type	VM Runtime Environments	Generation Option	Modifiable Object
Application COBOL program	<i>applname.CBL</i>	COBOL or COBSQL	VM CMS and VM batch	None	No
Sample runtime EXEC	<i>applname.RXX</i>	EXECX	VM CMS and VM batch	/RUNFILE	Review and possible modification required

Table 66. Objects Produced for Tables by the Generation Facility for VM

Object Type	OS/2 File Name and Extension	VM File Type	VM Runtime Environments	Generation Option	Modifiable Object
Table COBOL program	<i>tablname.CBL</i>	COBOL	VM CMS and VM batch	/GENTABLES	No

Table 67. Objects Produced for Map Groups by the Generation Facility for VM

Object Type	OS/2 File Name and Extension	VM File Type	VM Runtime Environments	Generation Option	Modifiable Object
Map group format module (See note 1)	<i>mapgnameFM.FMT</i>	EZEFOBJ	VM CMS	/GENMAPS /GENHELPMAPS	No
Online print services program (See note 2)	<i>mapgname.CBL</i>	COBOL	VM CMS	/GENMAPS /GENHELPMAPS	No
Batch print services program (See note 2)	<i>mapgnameP1.CBL</i>	COBOL	VM batch	/GENMAPS /GENHELPMAPS	No

Table 68. Objects Produced for All Part Types by the Generation Facility for VM (Programs, Tables, Map Groups)

Object Type	OS/2 File Name and Extension	VM File Type	VM Runtime Environments	Generation Option	Modifiable Object
Listing File (See note 3)	<i>mbrname.LST</i>	N/A	VM CMS and VM batch	/LISTING /LISTINGONERROR	No
Parts Control File (See note 4)	<i>mbrname.PCF</i>	EZEPCLT	VM CMS and VM batch	None	No
Preparation Command File (See note 3)	<i>mbrname.PRP</i>	N/A	VM CMS and VM batch	/PREPFILE	No
Preparation REXX exec	<i>applname.RXP</i>	EXECP	VM CMS and VM batch	/PREPFILE	Yes

Notes:

1. This object is produced only if the map group contains terminal maps.
2. This object is produced only if the map group contains print maps.
3. These files are not transferred to the host. The OS/2 file name is the same as the name of the part specified on the GENERATE subcommand.
4. One parts control file is created for a GENERATE subcommand. The parts control file has the same name as the part specified on the GENERATE subcommand. The parts control file for the program lists all parts generated for the program and all parts generated for the associated map groups and tables.

Objects Generated for Programs

Application COBOL Program

The generated program is a COBOL program that contains the following:

- Program control logic
- Logic for program processes, statement groups, and I/O operations
- Data for both the program and program control

The program control logic performs the following functions for a program, as needed:

- Initialization
- Cleanup at end of program
- Error reporting
- Segmentation support, including environment saving and restoration
- Transfer of control

Sample Runtime REXX exec

The VisualAge Generator generation facility produces a sample runtime REXX exec for running programs in the VM CMS and VM batch environments when the /RUNFILE generation option is specified; /RUNFILE is the default. The generated runtime REXX exec might not be complete, and must be reviewed and modified if necessary, before it is used. Refer to “Chapter 51. Modifying the Sample Runtime REXX Execs” on page 497 for more information.

Objects Generated for Tables

Table COBOL Program

The table program is a COBOL program that contains the table contents defined in program working storage. This object is produced when you specify the /GENTABLES generation option; /GENTABLES is the default. This allows tables to be generated independently of programs when the contents of a table needs to be changed.

Objects Generated for Map Groups

Online Print Services Program

The online print services COBOL program is generated for map groups that contain print maps. The online print services program is a COBOL program that performs print I/O, output formatting, and SET map CLEAR and EMPTY functions for print maps in the VM CMS environment. This object is produced when you specify the /GENMAPS or /GENHELPMAPS generation options (which are the defaults) during program generation.

Batch Print Services Program

The batch print services program is a COBOL program that formats data for line printers and writes the data to the printer output file (directly to the printer or a QSAM file). The batch print services program performs print I/O, output formatting, and SET map CLEAR and EMPTY functions for print maps. This program is used with programs that run in the VM batch environment. This object is produced when you specify the /GENMAPS or the /GENHELPMAPS generation options, which are the defaults.

Map Group Format Module

The map group format module is a generated structure that describes the map layout for terminal maps in the map group. The generator builds the structure as an object module for the VM CMS environment. The map group format module is produced only if the map group contains terminal maps. This

object is produced when you specify the /GENMAPS or the /GENHELPMAPS generation options, which are the defaults.

Objects Generated for All Part Types

The following lists the objects generated for part types (programs, tables, and map groups).

Listing File

A source listing of the part being generated and its associates is produced if one of the print options (/LISTING or /LISTINGONERROR) is specified during generation. The /NOLISTING print option is the default. The listing includes any messages that were issued and options that were in effect during generation.

Parts Control File

The parts control file is a list of all objects produced by the generator the last time generation was performed for the program, table, or map group with the same name as the parts control file part. This list provides a record of generated objects.

Preparation Command File

A preparation command file is generated when the /PREPFILE generation option, which is the default, is specified. The preparation command file produced by a GENERATE subcommand contains logic to prepare the generated parts. If the generated program includes tables and maps, the logic for preparing them is contained in the preparation command file.

The preparation command file runs on the workstation and contains commands used to transfer the generated files to the target host system. One preparation command file is created for each GENERATE subcommand that was issued.

Preparation REXX exec

When using VisualAge Generator, a preparation REXX exec file (*mbrname.RXP*) is generated when the /PREPFILE generation option, which is the default, is specified.

The VisualAge Generator program generator produces a REXX exec that contains commands to start the appropriate precompile, compile and linkage editor steps for the generated programs.

- For COBOL programs, the preparation REXX exec will compile and link-edit the program.

- For COBOL programs that use DB2/VSE, the preparation REXX exec will DB2/VSE precompile, COBOL compile and link-edit the program.
- For COBOL program map group format modules, the preparation REXX exec will link-edit the module.

Only one preparation REXX exec file is created for each GENERATE subcommand issued, even if the GENERATE subcommand generates multiple parts. The REXX exec file is produced from templates that you can modify to meet your requirements for preparation.

Refer to the *VisualAge Generator Generation Guide* for information about modifying the templates.

Chapter 50. Preparing Generated Programs on VM Systems

This chapter describes preparation procedures that are common to all runtime environments. These preparation procedures include the following steps:

- Running the preparation job
- Analyzing the preparation listing
- Preparing and running an SQL program.

Running the Preparation REXX Exec

VisualAge Generator automatically runs the preparation REXX exec built by the generator if the /PREP generation option, which is the default, was specified. If preparation ends with an error that can be corrected without requiring the program to be generated again, you can run the exec again by entering the CMS command:

```
PIPE REXX (appl EXEC)
```

Analyzing the Preparation Listing

If you have errors in the generated preparation REXX exec, check the generation listing for error messages related to the generation of the exec. Errors can occur if the templates used to create the preparation REXX exec are not modified correctly. Refer to the *VisualAge Generator Generation Guide* for information about modifying the templates.

Analyzing the SQL Precompile Step

The precompile step is successful if the return code is 4 or less.

You will not usually receive any messages from the SQL precompiler. If you do, refer to “DB2/VSE Precompiler Messages” on page 247. If you cannot determine the cause of the error from this manual, refer to the appropriate SQL messages manual or have your system administrator contact the IBM Support Center.

Analyzing the Compile Step

The compile step is successful if the return code is 4 or less.

Several common compiler messages are possible if you do any of the following:

- Use SQL row records
- Use the OPTIMIZE compile option
- Assign or move a data item to itself
- Use /MATH=COBOL

These common messages can be found in “COBOL Compilation Messages” on page 561 . If you encounter COBOL messages other than those found in that section, check the listing file from the generation for any error messages. Correct these errors and then generate and prepare the program again. If there are no errors in the listing file from the generation, give the COBOL compile listing that contains the error and the associated COBOL code to your system administrator and have the administrator contact the IBM Support Center.

Analyzing the Link-Edit Step

This section discusses how to resolve link-edit errors.

Link-Edit Errors Related to Missing Modules

Some modules from products such as SQL and VisualAge Generator Server must be included in the load module. A small set of VisualAge Generator Server functions are linked with each generated COBOL program. The VisualAge Generator Server modules are included with one or more linkage editor statements in the following form:

```
INCLUDE SELALMD(modulename)
```

If you generated your program to use dynamic linkage for calls and transfers to other programs, then the link-edit step should always end with a return code of 0. If you receive a nonzero return code, review your link-edit listing for “module not found” messages.

The link-edit step may end with a nonzero return code if you generated your program to use calls to another program and have defined a linkage editor control statement file (file name *part.LED*) for generation. The linkage editor control statement file is used by the generator as link-edit input for linking the program. The link-edit step ends with a return code of 8 until all the programs to be linked together have been generated and prepared. If all the programs have been prepared, the link step should give a return code of 0, indicating that the program is ready to run. Refer to the *VisualAge Generator Generation Guide* for additional information on specifying the linkage editor control statements for statically linked modules.

Other Link-Edit Errors

If you receive a nonzero return code for a reason other than a module not being found in the initial link-edit step, then refer to the LKED command in the *VM/ESA CMS Command Reference*. If the link-edit step is not successful using the linkage editor control statements you provided, review the information about specifying linkage editor control statements for static linkage in the *VisualAge Generator Generation Guide*.

Preparing to Run an SQL Program

This section describes the tasks that need to be done to enable program users to run an SQL program, such as granting authority and defining synonyms for SQL tables.

Granting Authority to Program Users

When you have finished preparing your generated program, you need to grant authority to the program users so they can run the program using DB2/VSE. You can do this by using the following GRANT command:

```
GRANT RUN ON myappl TO user1, user2, user3
```

If the KEEP parameter (the default) is specified in the DB2/VSE precompile for a program, you need to grant authority one time for each program user. If the REVOKE parameter is not specified, you must grant authority each time the program is precompiled.

Defining Synonyms for SQL Tables

Depending on how you specified the table when you defined your SQL row records in the program, you might need to create a synonym for the table before you can run the program. If you do not fully qualify a table name in an SQL row record with the table creator identifier, the database manager supplies a default qualifier.

For static SQL statements within your program, the default qualifier is the DB2/VSE authorization ID that is used during program preparation. Refer to the appropriate DB2/VSE manual for more information. If you are using dynamic SQL statements within your program, the default qualifier is the runtime authorization ID. All VisualAge Generator programs run in static mode. If you use table name host variables or runtime statement build, individual SQL statements use dynamic mode.

If DB2/VSE synonyms are used for table names, the program mode determines who must create the synonyms. If the SQL row record is used with dynamic SQL statements in a generated program, each program user

must create the synonym for the table. If the SQL row record is used only in static SQL statements in a generated program, only the programmer identified in the precompile operation needs to create a synonym for the table. If you specified a fully qualified table name when you defined your SQL row record, then you do not need to do anything special when you precompile the program.

For additional information on using unqualified table names and synonyms, refer to the *VisualAge Generator Design Guide*.

Using LE Runtime Options for VM CMS and VM batch

VisualAge Generator programs require certain LE runtime options that might not be the defaults for your LE installation. Because of the manner in which a VisualAge Generator program is started, LE runtime options for COBOL program programs must be specified in the CSECT CEEUOPT. CEEUOPT is used to resolve a weak external reference when the generated COBOL program is linked.

The *LE Programming Guide* explains how to create a program defaults module, CEEUOPT, that can be linked to a program to set runtime options for that program. The LE product ships a sample file called CEEUOPT ASSEMBLE that can be used to override the LE product or installation defaults (CSECT CEEDOPT). VisualAge Generator Server for MVS, VSE, and VM ships a modified version of CEEUOPT that is called ELAEUOPT ASSEMBLE. ELAEUOPT is included in some modules in the VisualAge Generator Server for MVS, VSE, and VM LOADLIB and is also included in VisualAge Generator programs that you create. It contains runtime options that are necessary for the VM environment. In most cases, the LE runtime options that shipped in ELAEUOPT are sufficient.

Only one of the runtime options specified in ELAEUOPT is required by VisualAge Generator Server for MVS, VSE, and VM. It is STORAGE(00,...) and indicates that LE should initialize COBOL working storage to binary zeroes. The other runtime options are ones that are recommended by the COBOL and LE documentation for COBOL programs. Refer to the following manuals for information about LE runtime options:

- *COBOL Programming Guide* (SC26-4767)
- *COBOL Compiler and Run-Time Migration Guide* (GC26-4764)
- *LE Programming Reference* (SC26-3312)
- *LE Run-Time Migration Guide* (SC26-8232)

The runtime options that are specified in the ELAEUOPT that is shipped with VisualAge Generator Server for MVS, VSE, and VM are:


```
CEEXOPT
  ABTERMENC=(ABEND),
  ALL31=(OFF),
  CBLQDA=(OFF),
  STACK=(64K,64K,BELOW,KEEP),
  STORAGE=(00,NONE,NONE,8K),
  TERMTHDACT=(DUMP),
  TRACE=(OFF,4K,DUMP,LE=0),
  TRAP=(ON)
```

The actual runtime options in the ELAEUOPT TEXT file on your system may be different if the person who installed VisualAge Generator Server for MVS, VSE, and VM changed or added any options. Contact that person for a copy of the ELAEUOPT source file.

The link-edit templates that are shipped with VisualAge Generator have a statement that includes ELAEUOPT. If ELAPREP determines you are using COBOL, ELAPREP will include the ELAEUOPT TEXT file in the generated program. If COBOL is being used, the INCLUDE ELAEUOPT statement in the link-edit file will have no effect because ELAPREP will set the FILEDEF for ELAEUOPT to DUMMY.

If you want to change or add additional LE runtime options, you can do this in one of the following ways:

- Ask the person who installed VisualAge Generator Server for MVS, VSE, and VM to obtain a copy of the ELAEUOPT source file. Modify it, assemble it, and store the resulting text file on a file mode that is accessed when the preparation REXX exec is run. When you run your preparation REXX exec, ELAPREP is called and will access the VisualAge Generator Server for MVS, VSE, and VM disk at the first available file mode in the CMS search order. Because the ELAEUOPT TEXT file already exists on that disk, you must ensure that the file mode containing your copy of the ELAEUOPT TEXT will be accessed before the VisualAge Generator Server for MVS, VSE, and VM disk in the CMS search order.
- If you think everyone at your installation will want your runtime options, ask the person who installed VisualAge Generator Server for MVS, VSE, and VM for a local modification to be made to ELAEUOPT using VMSES/E.
- A third approach is similar to the first approach, but your text deck will have a file name that is not ELAEUOPT. Modify the link-edit templates that are shipped with VisualAge Generator to include your text deck instead of ELAEUOPT. Modify the preparation REXX exec to specify the USEREXECL= option on the call to ELAPREP. USEREXECL= specifies the name of an exec to call just before the link-edit step. This exec should issue a CMS FILEDEF command to associate the include in the link-edit template to your text deck.

The last approach allows you to create several versions of CEEUOPT with different runtime options. You can then include a specific version of CEEUOPT with a particular VisualAge Generator program as required.

Chapter 51. Modifying the Sample Runtime REXX Execs

This chapter contains the information you need to modify the sample runtime REXX execs that were created during program generation. If you specified /RUNFILE at generation, the generator created a sample runtime REXX exec for running the generated program.

The generated sample runtime REXX exec is based on templates supplied with VisualAge Generator. It is also based on the VisualAge Generator program definition and the resource association file that was specified during the generation process. Because the generated sample runtime REXX exec might not be complete, it must be reviewed, and modified if necessary, before it is used. You might need to modify the sample runtime REXX execs for the following reasons:

- The generator does not include FILEDEF or DLBL commands to allocate data sets accessed when an allocation moves an alternate DDNAME to EZEDEST or EZEDESTP.
- The generator does not include FILEDEF or DLBL commands in the exec to allocate data sets accessed by programs that were transferred-to using the EZE word EZEAPP.
- You might want to tailor the file allocations for the required or optional files described in “Required and Optional File Descriptions” on page 475.

When the sample runtime REXX execs are transferred to the VM system, they will have a file type of EXECX. You must change the file type to EXEC for all of the sample runtime REXX execs that are generated. This must be done to allow the runtime exec of the first program in the run unit to be started by just entering the exec file name. It also allows the runtime execs of called or transferred-to programs to be invoked by the runtime exec of the first program.

Before you invoke the main runtime REXX exec, review and tailor all runtime REXX execs for that run unit.

If you get an error while using the runtime REXX exec, check the generation listing for the programs involved for any error messages related to sample runtime REXX exec generation. In addition, ensure the tailoring that was done for the templates is correct. Also check any changes you made when you customized the sample runtime REXX exec.

Tailoring Sample REXX Exec Templates Before Generation

VisualAge Generator creates a sample runtime REXX exec for both VM CMS and VM batch environments. The sample runtime REXX exec is based on templates supplied with VisualAge Generator. The default templates shipped with VisualAge Generator are installed in the TEMPLATE subdirectory of the VisualAge Generator product directory.

Your system administrator can tailor what is generated into the sample runtime REXX execs by changing the templates prior to generation. If you always need to make the same changes to the sample runtime execs, tailoring the templates can help reduce the number of manual changes required after the sample runtime exec generation. Refer to the /TEMPLATES generation option in the *VisualAge Generator Generation Guide* for more information about how to use customized templates.

Some of the reasons to tailor the templates include the following:

- Adding calls to other execs or non-VisualAge Generator programs before starting the VisualAge Generator program
- Adding FILEDEF statements to override definitions for required files described in “Required and Optional File Descriptions” on page 475
- Adding FILEDEF statements for optional files described in “Required and Optional File Descriptions” on page 475

Table 69 shows the base templates used for generating the sample runtime REXX execs. The templates are selected based on whether the program is a main or called program.

Table 69. Base Templates Used for Generating Sample Runtime REXX Execs

Program	Templates	EXECs Called by Template
Main	EFK2CERB	ELASETUP ELACLEAN ELARUN
Called	EFK2CERA	None

If your VisualAge Generator program used files for input or output, additional templates will be included in the sample runtime REXX exec that is generated. Templates EFK2CERA and EFK2CERB contain an ?ALLOC? line that indicates where the generation function is to insert copies of templates from Table 70 on page 499. These templates insert calls to ELACALLC EXEC to do FILEDEF and DLBL commands for files used by the program.

The templates that are included in the sample runtime exec for file allocation based on the file access method and whether the record is input or output are shown in Table 70.

Table 70. Templates for Generating File Allocation Statements

File Access Method	Input or Output	Possible Record Organization	Template Name
VSAM, VSAMRS	Input	Indexed Relative Serial	EFK2CVVI
VSAM, VSAMRS	Output	Indexed Relative Serial	EFK2CVVO
SEQ, SEQRS	Input	Serial	EFK2CVSI
SEQ, SEQRS	Output	Serial	EFK2CVSO

Note: The statements generated from file allocation templates are included at points marked by ?ALLOC? line in the base template.

If your VisualAge Generator program called or transferred to other VisualAge Generator programs or used EZEDEST or EZEDESTP, additional templates will be included in the generated sample runtime REXX exec. Templates EFK2CERA and EFK2CERB contain an ?ALLOC? line that indicates where the generation function is to insert copies of templates from Table 71. These templates insert a call to the runtime exec for known programs or insert comments indicating where you will need to insert code to handle a special situation.

The templates that are included in the sample runtime exec for special file allocation situations are shown in Table 71.

Table 71. Templates for Special File Allocation Situations

Template Name	Function	Exec Procedure Called
EFK2CVCL	Allocate files for known transferred-to or called programs	Runtime exec for transferred-to or called program
EFK2CVEA	Comment indicating where to insert commands to run the exec for programs transferred-to using EZEAPP	None
EFK2CVED	Comment indicating where to insert FILEDEF or DLBL commands for data sets using EZEDEST or EZEDESTP	None

Note: The statements generated from special file allocation situations are included at points marked by ?ALLOC? line in the base template.

Modifying the Sample Runtime REXX Execs After Generation

The generated sample runtime REXX exec might not be complete and must be reviewed, and modified if necessary, before it is used. This section describes items that would cause you to modify the generated sample runtime REXX exec.

Note: Do not directly modify the runtime REXX exec that VisualAge Generator produces in TeamConnection because the build that produced them would be out of date; the next build replaces any changes that you make. Either make the changes on the host when you move the runtime REXX exec to the host, or copy the generated REXX exec to another file in TeamConnection that would be transferred to the host for installation.

File Allocation Considerations

If the VisualAge Generator program requires files for input or output, these files must be defined before the ELARUN EXEC is called to run the first program in the run unit. This section describes items related to file allocation that would cause you to modify the generated sample runtime REXX exec.

General File Allocation Considerations

If your VisualAge Generator program uses files for input or output, the runtime exec should be modified to link and access the necessary minidisks or SFS directories that contain the files. If these are VSAM files, a DLBL command for the master catalog should be added to the runtime exec. Also, ASSGN commands for the master catalog and VSAM files may need to be added to the runtime exec.

Note: The DOS environment must be set ON while the ASSGN commands are issued.

Files Used By The Main Program

If your VisualAge Generator program used files for input or output, the sample runtime REXX exec will include templates, which are described in Table 70 on page 499. These templates insert a call to the ELACALLC EXEC to do FILEDEF or DLBL commands for files used by the program. The parameters for ELACALLC are generated based on the information found in the Resource Association file that was specified when the program was generated. The parameters on the calls to ELACALLC must be verified and modified if necessary. Refer to the online ELACALLC documentation (enter HELP ELA) or the documentation in “ELACALLC EXEC” on page 518.

Using EZEDEST or EZEDESTP for File Allocation

If your VisualAge Generator program uses the EZE words EZEDEST or EZEDESTP to identify files to use for input or output, the sample runtime REXX exec will include template EFK2CVED, which is described in Table 71 on page 499. This template inserts comments indicating where you might need to insert FILEDEF or DLBL commands for files using the EZEDEST or EZEDESTP words. If your program moves an alternate DDNAME to EZEDEST or EZEDESTP, you must add FILEDEF or DLBL commands for the files that you wish to associate with the DDNAME. However, if your program moves a data set name to EZEDEST or EZEDESTP, you do not need to add FILEDEF or DLBL commands for these files.

Files Used by Other Known Programs

If your VisualAge Generator program calls other VisualAge Generator programs or explicitly transfers to other VisualAge Generator programs by a DXFR or an XFER statement, the sample runtime REXX exec will include template EFK2CVCL, which is described in Table 71 on page 499. This template inserts commands to run the runtime exec for VisualAge Generator programs. These runtime execs are called in such a way that it results only in the allocation of any data sets needed by the called or transferred-to program.

If your VisualAge Generator program calls non-VisualAge Generator programs or explicitly transfers to non-VisualAge Generator programs by a DXFR or an XFER statement, template EFK2CVCL is included. Just as described above, this template inserts commands to run an exec that is the same name as the non-VisualAge Generator program. This exec is expected to perform any file allocations that are necessary for the non-VisualAge Generator program. If your program requires any file allocations, you must either create the exec to perform this function or delete the call and add the appropriate FILEDEF or DLBL commands at this point. If your program does not require any file allocations, you must delete the call to the exec.

Files Used By EZEAPP Programs

If your VisualAge Generator program uses the EZE word EZEAPP with XFER and DXFR statements, the sample runtime REXX exec will include template EFK2CVEA, which is described in Table 71 on page 499. This template inserts comments indicating where you might need to add a call to the runtime exec of the transferred-to program. Running that runtime exec is necessary if that program uses additional files for input or output. That runtime exec should be called so that only file allocations are done. For example, to call the runtime exec of program APPLB so that just file allocations are done, insert the following line at the location indicated by template EFK2VCEA:

```
CALL EXEC_CALL_APP '%EZEAPP%' ELA_ACTION
```

If you use EZEAPP to refer to a non-VisualAge Generator program, then template EFK2CVCL is included. Just as described above, this template inserts comments indicating where you might want to add a call to an exec to perform file allocations for your non-VisualAge Generator program. Alternatively, you could add the appropriate FILEDEF or DLBL commands at this point. If your program does not require any file allocations, no action is required.

Required and Optional File Allocations

You can also customize the sample runtime exec to allocate any of the required and optional file allocations that are described in “Required and Optional File Descriptions” on page 475. Normally, the ELARUN EXEC will allocate the required files before starting your program. But if a FILEDEF command is specified for one of these files in your runtime exec, the ELARUN EXEC uses your definition for that file. For example, you could code a FILEDEF command in your runtime exec so that a snap dump is not generated by adding the following statement to your runtime exec just before the call to ELARUN:

```
'FILEDEF ELASNAP DUMMY'
```

SQL Considerations

If the VisualAge Generator program requires SQL databases, some tailoring of the generated sample runtime REXX exec may be necessary. Refer to “Database Considerations” on page 472 for changes that might be necessary to the runtime exec.

Runtime REXX EXEC Examples

The examples referred to in this section were generated from the templates shipped with VisualAge Generator. Your generated sample REXX exec might be different if your administrator has tailored the templates to reflect the standards of your organization.

A Main Program: Figure 83 on page 503 shows a sample runtime exec that was generated to invoke a main transaction program. This program calls program A21VS19 shown in Figure 84 on page 507


```

/*REXX*/
/*
=====
      EFK2CERB - EXECUTE VM APPLICATION

      MAIN APPLICATION: A21VS19
      TARGET ENVIRONMENT: VMCMS
      GENERATION DATE: 08/19/96
      GENERATION TIME: 09:36:00
      SQL USED: N
=====
*/
ADDRESS COMMAND
PARSE ARG INVAR$ '$$L$A$V$A$R$$:' ELAVARS
PARSE VAR ELAVARS ELA_APPL_FUNCTION ',' ELA_APPS ',' ,
              ELA_APPL_VMLoadLIB ',' ,
              ELA_SQL_USED ',' ELA_SQL_SID ',' ELA_SQL_SDB ',' ,
              ELA_DBCS ',' ELA_SQL_MODE

PARSE SOURCE . . ELA_EXEC_FN .
PARSE UPPER VAR INVAR$ . 'SID=' IN_SID .
PARSE UPPER VAR INVAR$ . 'SDB=' IN_SDB .
EZE_APPL_VMLoadLIB = 'EZEAPPL'
EZE_DBCS = 'N'
EZE_SQL = 'N'
EZE_SQL_MODE = 'MULTIUSER'
IF IN_SID = '' THEN
    EZE_SQL_SID = '
ELSE
    EZE_SQL_SID = IN_SID
IF IN_SDB = '' THEN
    EZE_SQL_SDB = 'SQLDB7'
ELSE
    EZE_SQL_SDB = IN_SDB

/* IS APPL NAME IN VERIFICATION STRING? */
/* IF NAME FOUND, REXX EXEC ALREADY RAN ONCE */
IF WORDPOS(ELA_EXEC_FN,ELA_APPS) > 0 THEN
    EXIT 4 /* EXIT IF ALREADY BEEN HERE */
ELSE
    ELA_APPS = ELA_APPS ELA_EXEC_FN /* ADD APPL NAME TO LIST */

IF WORDPOS(EZE_APPL_VMLoadLIB,ELA_APPL_VMLoadLIB) = 0 THEN
    ELA_APPL_VMLoadLIB = ELA_APPL_VMLoadLIB EZE_APPL_VMLoadLIB

IF ELA_DBCS <> 'Y' & ELA_DBCS <> 'N' THEN ELA_DBCS = 'N'
IF EZE_DBCS = 'Y' THEN ELA_DBCS = EZE_DBCS

IF ELA_SQL_MODE <> 'MULTIUSER' & ELA_SQL_MODE <> 'SINGLEUSER' THEN
    ELA_SQL_MODE = 'MULTIUSER'
IF EZE_SQL_MODE = 'SINGLEUSER' THEN ELA_SQL_MODE = EZE_SQL_MODE

```

Figure 83. REXX Exec for Main Program (Part 1 of 4)

```

IF ELA_SQL_USED <> 'Y' & ELA_SQL_USED <> 'N' THEN
  ELA_SQL_USED = 'N'
IF EZE_SQL = 'Y' THEN
  ELA_SQL_USED = EZE_SQL
SELECT
  WHEN EZE_SQL = 'Y' THEN DOFLAG = 'YES'
  WHEN IN_SID <> '' THEN DOFLAG = 'YES'
  WHEN IN_SDB <> '' THEN DOFLAG = 'YES'
  OTHERWISE DOFLAG = 'NO'
END
IF DOFLAG = 'YES' THEN DO
  IF WORDPOS(EZE_SQL_SID,ELA_SQL_SID) = 0 THEN
    ELA_SQL_SID = ELA_SQL_SID EZE_SQL_SID
  IF WORDPOS(EZE_SQL_SDB,ELA_SQL_SDB) = 0 THEN
    ELA_SQL_SDB = ELA_SQL_SDB EZE_SQL_SDB
  END
END

IF ELA_APPL_FUNCTION <> 'ALLOCATE' & ELA_APPL_FUNCTION <> 'FREE'
THEN
  'EXEC ELASETUP ALLOCATE'
ELSE DO
  CALL FILE_LIST ELA_APPL_FUNCTION
  'PIPE (STAGESEP |) VAR ELA_APPS | VAR ELA_APPS 1'
  IF ELA_APPL_FUNCTION = 'ALLOCATE' THEN DO
    'PIPE (STAGESEP |) VAR ELA_APPL_VMLoadLIB ',
    ' | VAR ELA_APPL_VMLoadLIB 1'
    'PIPE (STAGESEP |) VAR ELA_SQL_USED | VAR ELA_SQL_USED 1'
    'PIPE (STAGESEP |) VAR ELA_SQL_SID | VAR ELA_SQL_SID 1'
    'PIPE (STAGESEP |) VAR ELA_SQL_SDB | VAR ELA_SQL_SDB 1'
    'PIPE (STAGESEP |) VAR ELA_DBCS | VAR ELA_DBCS 1'
    'PIPE (STAGESEP |) VAR ELA_SQL_MODE | VAR ELA_SQL_MODE 1'
  END
  EXIT 0
END

CALL FILE_LIST 'ALLOCATE'

/* DEFINE MASTER CATALOG AND DO ASSGN FOR VSAM FILES */
'SET DOS ON'
'ASSGN SYSCAT F'
'ASSGN SYS001 F'
'SET DOS OFF'
'DLBL IJSYSCT F DSN MASTCAT ( VSAM PERM )'

'EXEC ELARUN A21VS19,'ELA_APPL_VMLoadLIB','',
  ELA_SQL_USED','ELA_SQL_SID','ELA_SQL_SDB','',
  ELA_DBCS','ELA_SQL_MODE
ELA_EXEC_RC = RC

```

Figure 83. REXX Exec for Main Program (Part 2 of 4)

```

IF ELA_EXEC_RC <> 0 THEN
  'XMITMSG 440 "A21VS19" "'ELA_EXEC_RC'"' ,
  ' (APPLID ELA CALLER RUN HEADER ERRMSG'
ELA_APPS = ELA_EXEC_FN
CALL FILE_LIST 'FREE'
'EXEC ELACLEAN FREE'

EXIT ELA_EXEC_RC

/*****
/* FILE ALLOCATIONS FOR THIS APPLICATION, TRANSFERRED-TO APPLICATIONS */
/* AND CALLED APPLICATIONS. */
*****/
FILE_LIST: PROCEDURE,
  EXPOSE ELA_EXEC_FN ELA_APPS ELA_APPL_VMLoadLIB,
         ELA_SQL_USED ELA_SQL_SID ELA_SQL_SDB,
         ELA_SQL_MODE ELA_DBCS
ARG ELA_ACTION

/* EFK2CVVO - VSAM OUTPUT FILE ALLOCATION */
/* If a CP LINK or ACCESS of a minidisk is */
/* required, do it here. The filemode used by default by the */
/* ELACALLC EXEC will be "*". For less search overhead during */
/* application execution, set the assignment statement for the */
/* "OVRFM" variable below to the specific filemode. */
OVRFM = ' ' /* override filemode */
'EXEC ELACALLC 'ELA_ACTION',VSAM,WRITE,,RXVS09',
',CSPTSTV.RSB21VS9.VSAMI,'OVRFM

/* EFK2CVSI - QSAM INPUT FILE ALLOCATION */
/* If a CP LINK or ACCESS of a minidisk or SFS directory is */
/* required, do it here. If the variable "OVRFM" is set to ' ' */
/* (null), the filemode that was specified in the resource */
/* association file on the SYSNAME option will be used or "*" */
/* will be used if the resource association file had no */
/* filemode. To override a hardcoded filemode in the resource */
/* association file, set "OVRFM" to the proper filemode before */
/* the call to the ELACALLC EXEC. */
OVRFM = ' ' /* override filemode */
'EXEC ELACALLC 'ELA_ACTION',QSAM,READ,,RSVS07A',
',RSVS07 REPRO,'OVRFM

/* EFK2CVVO - VSAM OUTPUT FILE ALLOCATION */
/* If a CP LINK or ACCESS of a minidisk is */
/* required, do it here. The filemode used by default by the */
/* ELACALLC EXEC will be "*". For less search overhead during */
/* application execution, set the assignment statement for the */
/* "OVRFM" variable below to the specific filemode. */
OVRFM = ' ' /* override filemode */
'EXEC ELACALLC 'ELA_ACTION',VSAM,WRITE,,RSVS09',
',CSPTSTV.RSBVS09S.VSAMS,'OVRFM

```

Figure 83. REXX Exec for Main Program (Part 3 of 4)

```

/* EFK2CVCL - ALLOCATE OR FREE FILES FOR TRANSFERRED-TO */
/*           OR CALLED APPLICATION A21VS20           */
CALL EXEC_CALL_APP 'A21VS20' ELA_ACTION

RETURN 0

/*****
/* ROUTINE TO SUPPORT EXECUTING EXECS FOR CALLED APPLICATIONS. */
/*****
EXEC_CALL_APP: PROCEDURE,
    EXPOSE ELA_EXEC_FN ELA_APPS ELA_APPL_VMLoadLIB,
           ELA_SQL_USED ELA_SQL_SID ELA_SQL_SDB,
           ELA_SQL_MODE ELA_DBCS
ARG ELA_CALL_APP ELA_ACTION
IF WORDPOS(ELA_CALL_APP,ELA_APPS) = 0 THEN
    'EXEC' ELA_CALL_APP '$E$L$A$V$A$R$$:'ELA_ACTION','ELA_APPS','',
           ELA_APPL_VMLoadLIB','ELA_SQL_USED','ELA_SQL_SID','ELA_SQL_SDB','',
           ELA_DBCS','ELA_SQL_MODE
RETURN 0

```

Figure 83. REXX Exec for Main Program (Part 4 of 4)

A Called Program: Figure 84 on page 507 shows a sample runtime exec that was generated to invoke a called transaction program.

```

/*REXX*/
/*
=====
      EFK2CERA - EXECUTE VM APPLICATION

      CALLED APPLICATION: A21VS20
      TARGET ENVIRONMENT: VMCMS
      GENERATION DATE: 08/19/96
      GENERATION TIME: 09:38:37
      SQL USED: N
=====
*/
ADDRESS COMMAND
PARSE ARG INVAR$ '$$L$$V$$R$$:' ELAVARS
PARSE VAR ELAVARS ELA_APPL_FUNCTION ',' ELA_APPS ',' ,
              ELA_APPL_VMLoadLIB ',' ,
              ELA_SQL_USED ',' ELA_SQL_SID ',' ELA_SQL_SDB ',' ,
              ELA_DBCS ',' ELA_SQL_MODE
PARSE SOURCE . . ELA_EXEC_FN .
EZE_APPL_VMLoadLIB = 'EZEAPPL'
EZE_DBCS = 'N'
EZE_SQL = 'N'
EZE_SQL_MODE = 'MULTIUSER'
EZE_SQL_SID = '
EZE_SQL_SDB = 'SQLDB7'

/* IS APPL NAME IN VERIFICATION STRING? */
/* IF NAME FOUND, REXX EXEC ALREADY RAN ONCE */
IF WORDPOS(ELA_EXEC_FN,ELA_APPS) > 0 THEN
  EXIT 4 /* EXIT IF ALREADY BEEN HERE */
ELSE
  ELA_APPS = ELA_APPS ELA_EXEC_FN /* ADD APPL NAME TO LIST */

IF WORDPOS(EZE_APPL_VMLoadLIB,ELA_APPL_VMLoadLIB) = 0 THEN
  ELA_APPL_VMLoadLIB = ELA_APPL_VMLoadLIB EZE_APPL_VMLoadLIB

IF ELA_DBCS <> 'Y' & ELA_DBCS <> 'N' THEN ELA_DBCS = 'N'
IF EZE_DBCS = 'Y' THEN ELA_DBCS = EZE_DBCS

IF ELA_SQL_MODE <> 'MULTIUSER' & ELA_SQL_MODE <> 'SINGLEUSER' THEN
  ELA_SQL_MODE = 'MULTIUSER'
IF EZE_SQL_MODE = 'SINGLEUSER' THEN ELA_SQL_MODE = EZE_SQL_MODE

```

Figure 84. REXX Exec for Called Program (Part 1 of 3)

```

IF ELA_SQL_USED <> 'Y' & ELA_SQL_USED <> 'N' THEN
  ELA_SQL_USED = 'N'
IF EZE_SQL = 'Y' THEN DO
  ELA_SQL_USED = EZE_SQL
  IF WORDPOS(EZE_SQL_SID,ELA_SQL_SID) = 0 THEN
    ELA_SQL_SID = ELA_SQL_SID EZE_SQL_SID
  IF WORDPOS(EZE_SQL_SDB,ELA_SQL_SDB) = 0 THEN
    ELA_SQL_SDB = ELA_SQL_SDB EZE_SQL_SDB
END
IF ELA_APPL_FUNCTION <> 'ALLOCATE' & ELA_APPL_FUNCTION <> 'FREE'
THEN
  EXIT 8
ELSE DO
  CALL FILE_LIST ELA_APPL_FUNCTION
  'PIPE (STAGESEP |) VAR ELA_APPS | VAR ELA_APPS 1'
  IF ELA_APPL_FUNCTION = 'ALLOCATE' THEN DO
    'PIPE (STAGESEP |) VAR ELA_APPL_VMLoadLIB ',
    ' | VAR ELA_APPL_VMLoadLIB 1'
    'PIPE (STAGESEP |) VAR ELA_SQL_USED | VAR ELA_SQL_USED 1'
    'PIPE (STAGESEP |) VAR ELA_SQL_SID | VAR ELA_SQL_SID 1'
    'PIPE (STAGESEP |) VAR ELA_SQL_SDB | VAR ELA_SQL_SDB 1'
    'PIPE (STAGESEP |) VAR ELA_DBCS | VAR ELA_DBCS 1'
    'PIPE (STAGESEP |) VAR ELA_SQL_MODE | VAR ELA_SQL_MODE 1'
  END
END

EXIT 0

/*****
/* FILE ALLOCATIONS FOR THIS APPLICATION, TRANSFERRED-TO APPLICATIONS */
/* AND CALLED APPLICATIONS. */
/*****
FILE_LIST: PROCEDURE,
  EXPOSE ELA_EXEC_FN ELA_APPS ELA_APPL_VMLoadLIB,
          ELA_SQL_USED ELA_SQL_SID ELA_SQL_SDB ,
          ELA_SQL_MODE ELA_DBCS
ARG ELA_ACTION

/* EFK2CVVI - VSAM INPUT FILE ALLOCATION */
/* If a CP LINK or ACCESS of a minidisk is */
/* required, do it here. The filemode used by default by the */
/* ELACALLC EXEC will be "*". For less search overhead during */
/* application execution, set the assignment statement for the */
/* "OVRFM" variable below to the specific filemode. */
OVRFM = '' /* override filemode */
'EXEC ELACALLC 'ELA_ACTION',VSAM,READ,,RXVS09',
',CSPTSTV.RSB21VS9.VSAMI,'OVRFM

```

Figure 84. REXX Exec for Called Program (Part 2 of 3)

```

/* EFK2CVVI - VSAM INPUT FILE ALLOCATION */
/* If a CP LINK or ACCESS of a minidisk is */
/* required, do it here. The filemode used by default by the */
/* ELACALLC EXEC will be "*". For less search overhead during */
/* application execution, set the assignment statement for the */
/* "OVRFM" variable below to the specific filemode. */
OVRFM = ' ' /* override filemode */
'EXEC ELACALLC 'ELA_ACTION',VSAM,READ,,RSVS09',
      ',CSPTSTV.RSBVS09S.VSAMS,'OVRFM

RETURN 0

/*****/
/* ROUTINE TO SUPPORT EXECUTING EXECS FOR CALLED APPLICATIONS. */
/*****/
EXEC_CALL_APP: PROCEDURE,
  EXPOSE ELA_EXEC_FN ELA_APPS ELA_APPL_VMLOADLIB,
         ELA_SQL_USED ELA_SQL_SID ELA_SQL_SDB,
         ELA_SQL_MODE ELA_DBCS
ARG ELA_CALL_APP ELA_ACTION
IF WORDPOS(ELA_CALL_APP,ELA_APPS) = 0 THEN
  'EXEC' ELA_CALL_APP '$E$L$A$V$A$R$$:'ELA_ACTION','ELA_APPS','
        ELA_APPL_VMLOADLIB','ELA_SQL_USED','ELA_SQL_SID','ELA_SQL_SDB','
        ELA_DBCS','ELA_SQL_MODE
RETURN 0

```

Figure 84. REXX Exec for Called Program (Part 3 of 3)

Chapter 52. Running a Generated Program on VM Systems

This chapter describes the unique steps required to run a generated COBOL program in a VM environment:

- Making VisualAge Generator Server for MVS, VSE, and VM, COBOL, and the generated modules available
- Running main programs

For general information on preparing your program for the runtime environment, see Chapter 50. Preparing Generated Programs on VM Systems. For information on modifying runtime execs, see Chapter 51. Modifying the Sample Runtime REXX Execs.

Making VisualAge Generator Server for MVS, VSE, and VM, COBOL, and the Generated Modules Available

For VisualAge Generator programs, the VisualAge Generator Server for MVS, VSE, and VM and the COBOL runtime libraries must be made available. The ELASETUP EXEC that is supplied with VisualAge Generator Server for MVS, VSE, and VM is used by the runtime REXX exec to access the location where VisualAge Generator Server for MVS, VSE, and VM resides. ELASETUP is customized during VisualAge Generator Server for MVS, VSE, and VM installation and should be available on a commonly accessed disk or SFS directory. If you are using a VM batch facility, you must ensure the batch machine running the program has access to VisualAge Generator Server for MVS, VSE, and VM and the COBOL runtime libraries.

The runtime REXX exec calls the ELARUN EXEC to run the VisualAge Generator program. ELARUN will access the COBOL runtime libraries before the program is started.

If your generated program contains SQL processes, the ELARUN EXEC will access the SQL product disk and issue an SQLINIT command if a database name was provided. For information on specifying a database name, refer to “Connection Considerations for SQL Programs” on page 479.

Running Main Programs under VM

Prior to starting a generated program, the program user must log on to the VM CMS environment. Refer to the VM CMS documentation for information about logging on to VM.

Either a main transaction program or a main batch program generated for VM CMS can be started from the VM CMS command line. In addition, a main batch program can be initiated in a batch machine. Called programs can only be started by another VisualAge Generator program or by a non-VisualAge Generator program.

In order to use the runtime REXX exec, it must be on a minidisk or SFS directory that is accessed. In other words, it must be in the CMS search order.

When the sample runtime execs are transferred to the VM system, they will have a file type of EXECX. After they are modified as required, all runtime execs in the run unit must be renamed to a file type of EXEC. This allows the runtime exec to be started by just typing its file name.

If you get an error while using the runtime exec, check the generation listing for the programs involved for any error messages related to runtime exec generation. In addition, ensure the tailoring that was done for the exec templates is correct. Also check any changes you made when you customized the sample runtime REXX exec.

VM Saved Shared Segments

Most generated programs are compiled and linked as members in a simulated operating system load library on VM. To create a saved shared segment, use these steps to create a relocatable module from the generated program and load it into a saved segment.

1. Change the **erase_text_option** variable found in the generated EXECPC file from Y to N, as follows:
`erase_text_option='N'`

The N option means that the preparation process will not erase the intermediate TEXT decks that are used for generation.

Another way to modify this variable is to change the template used to create the EXECPC file. The name of the template is EFK2CPXP.TPL. The TEXT decks for the map group format modules can be created by copying the EZEF OBJ file to a TEXT file.

2. Link the TEXT decks into a loaded segment or a MODULE, following the instructions in the *VM/ESA CMS Application Development Guide*. To resolve the unresolved references during loading, use the following TXTLIBS:
 - SELALKED (provided with VisualAge Generator)
 - VMLIB (provided with VM/ESA)
 - SCEELKED (provided with IBM LE for MVS & VM)
 - TEXT decks for SQL, if the generated programs contain SQL
3. To move the TEXT or MODULE files to saved segments, follow the documentation for saved segment planning and administration in the *VM/ESA Planning and Administration* manual.
4. Making an initial program into a saved segment is not recommended because the program will not be able to make DXFR calls that use XCTL or XFER for the transfer. However, if you do use a saved segment as an initial program, the ELARUN exec and the execution exec need to be modified as follows:
 - a. Make a copy of the ELARUN exec under a different name.
 - b. Edit the copy and replace the 'OSRUN 'parm1 statement with the appropriate statement to execute a program that has been loaded as a saved segment.
 - c. Make appropriate changes to either the ELARUN exec or the execution exec to make sure the saved segment is loaded.
 - d. Modify the exec that executes ELARUN to point to your modified ELARUN exec.

The restrictions for using generated programs as saved segments are:

- If the saved segment is an initial program, you will not be able to call DXFR using XCTL or XFER to other programs.
- You will not be able to call DXFR using XCTL or XFER to a program that is loaded as a saved segment.

Chapter 53. Moving Prepared Programs to Other Systems from VM Systems

You might need to move a prepared program from one system to another. For example you might have the compiler on one host development machine but want to run the program on several production machines.

The COBOL and VisualAge Generator Server products on the production machine must be at the same maintenance level as, or a higher level than the development machine.

Moving Prepared Programs To Another VM System

If a program has been completely prepared on one system and you want to move the prepared program to another system, perform the following steps:

1. Copy the files associated with the program, print services, table programs, and map group format module to a minidisk or SFS directory on the other system.
2. Copy the runtime REXX exec for the programs to a minidisk or SFS directory on the other system. Modify the runtime REXX exec as necessary for the new system.
3. Use the DB2/VSE utilities to back up the DB2/VSE packages for SQL program programs on the development system and reload them in the DB2/VSE database on the new system. You may need to modify the program runtime REXX execs to connect to the DB2/VSE database on the new system. For more information on connecting to databases, refer to “Connection Considerations for SQL Programs” on page 479.
4. Define files and databases used by the program on the new system.

Maintaining Backup Copies of Production Libraries

Follow your installation-defined guidelines and procedures for making backup copies of production libraries. Having backup copies of production libraries enables you to return to the prior level of a program in case of errors.

You might also want to set up procedures for saving a copy of the program source in external or internal source format at the time a program is generated. Having a copy of the program source enables you to recreate production level libraries at any change level if necessary.

Chapter 54. Execs for Preparing and Running Programs on VM Systems

Several VM execs are called by the VisualAge Generator Server generated preparation and runtime execs. To better support your process, you can modify these execs as required. The execs are:

Exec Purpose

ELACALLC

Allocate or deallocate program files

ELAPREP

Prepare a program

ELARUN

Run a program

ELASQLAA

Run the SQL preprocessor for a generated program

These execs are described in more detail in the following sections.

ELACALLC EXEC

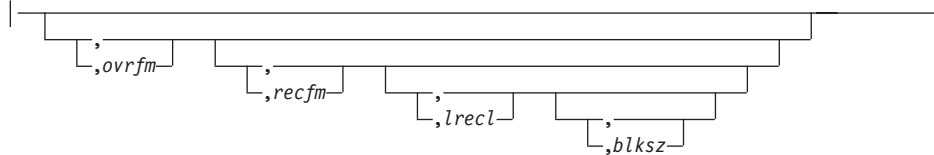
Purpose

Use the ELACALLC EXEC to allocate and deallocate files that are defined in the resource association file for a VM environment.

Format

→ ELACALLC (1) (2) *action,accmeth,inout,disp,ddn,dsn* Options | →

Options



Notes:

1. Do not specify nulls or blank spaces between any of the required or optional parameters. However, blank spaces are allowed within the data set name (*dsn*) parameter to separate the file name, file type, and file mode if it is a CMS file.
2. All parameters are positional with commas used as separators.

Parameters

action

The action, ALLOCATE or FREE, that you want to occur

accmeth

Access method (QSAM/VSAM)

inout

Open for input/output (READ/WRITE)

disp

DISP option for FILEDEF (OLD/MOD)

ddn

System resource name (DDNAME)

dsn

Data set name. Blank spaces are allowed within the data set name parameter to separate the file name and file type, and file mode if it is a CMS file.

Defaults

ovrfm

Override file mode; this is an optional parameter. If it is not specified, the file mode in the *dsn* parameter is used. If there is also no file mode in the *dsn*, then the default file mode is '*'.

recfm

Record format; an optional parameter

lrecl

Logical record length; an optional parameter

blksz

Blocksize; an optional parameter

Results

The proper FILEDEF (non-VSAM sequential files) or DLBL (VSAM files) command is issued to associate the file or data set with the specified system resource name (DDNAME) or to clear the file definition.

Usage

1. The required and optional parameters are positional, and must be entered as one string, separated by commas.
2. No input is required for optional parameters unless subsequent optional parameters are specified. Commas without values specified must be used to allow one optional parameter to assume the default if the next optional parameter in the positional order is to be entered with a value specified.
3. If an optional parameter is not specified or a comma is used as a placeholder for an optional parameter, the default value of the parameter for the DLBL or FILEDEF command will be used. For more information about these commands, refer to *VM/ESA: CMS Command Reference*

Examples

1. ELACALLC issues a FILEDEF command for a sequential file (QSAM) that will be opened for read and write (input and output):

```
ELACALLC ALLOCATE,QSAM,WRITE,OLD,FILEDD,USER1 FILE A,B
```

The data set name (file name and file type) is USER1 FILE and the resource association file indicated that it is on file mode A. However, the *ovrfm* option was used to override the file mode to change it to file mode B. The OLD operand causes the FILEDEF command to be issued with the DISP OLD option, which causes an existing file with this data set name to be replaced if the program writes to it. The *recfm*, *lrecl*, and *blksz* were not specified and will not be included in the FILEDEF command that is issued to allocate this file.

2. ELACALLC issues a FILEDEF command for a sequential file (QSAM) that will be opened for read and write (input and output):

```
ELACALLC ALLOCATE, QSAM, WRITE, MOD, FILEDD, USER2 FILE, , F, 80, 80
```

The data set name (file name and file type) specified is USER2 FILE. The MOD operand causes the FILEDEF command to be issued with the DISP MOD option, which causes records written to the file to be appended to the end of an existing file with this name. No file mode was specified in the data set name. The file mode will default to "*" because the *ovrfm* option was not specified. The *ovrfm* option was coded with two commas with no blank spaces between them because the subsequent *recfm* option was specified and the options are positional. The *recfm* (record format) option was specified to be fixed, and the *lrecl* (logical record length) and *blksz* (block size) options were both specified to be 80.

3. ELACALLC issues a DLBL command for a VSAM file definition:

```
ELACALLC ALLOCATE, VSAM, WRITE, OLD, FILEDD, USER3.DATA, C
```

The data set name is USER3.DATA and the file mode to be used in the DLBL command is specified as C by the *ovrfm* option.

4. ELACALLC issues a DLBL command for a VSAM file definition:

```
ELACALLC ALLOCATE, VSAM, WRITE, OLD, FILEDD, USER4.DATA.FILE
```

The data set name is USER4.DATA.FILE. The *ovrfm* option was not specified. Therefore, ELACALLC EXEC will look for the first DOS disk in the CMS search order and use that as the file mode in the DLBL command.

ELAPREP EXEC

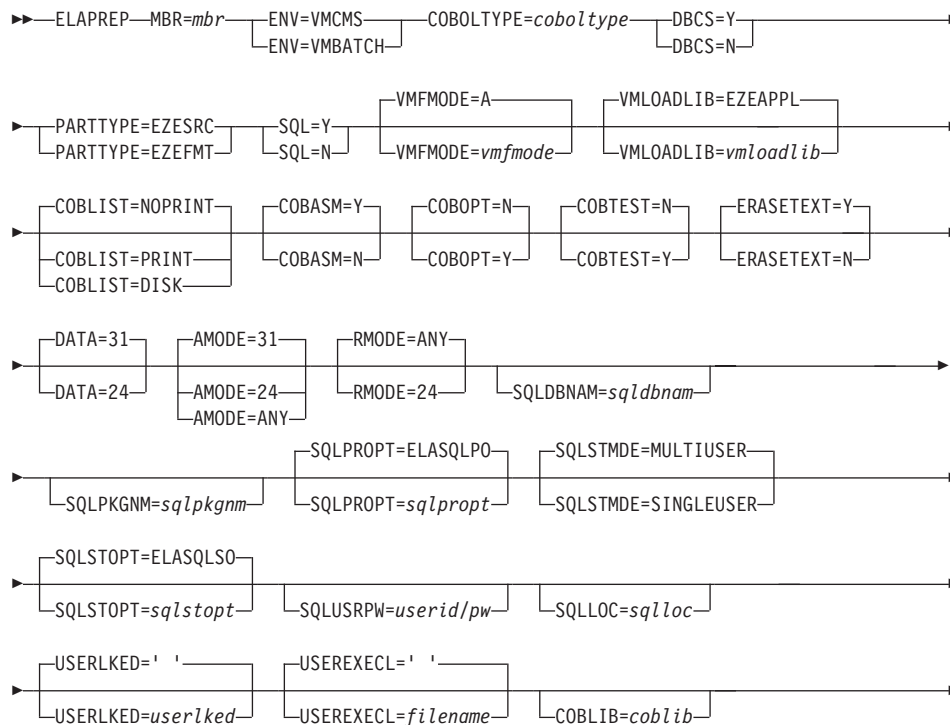
Purpose

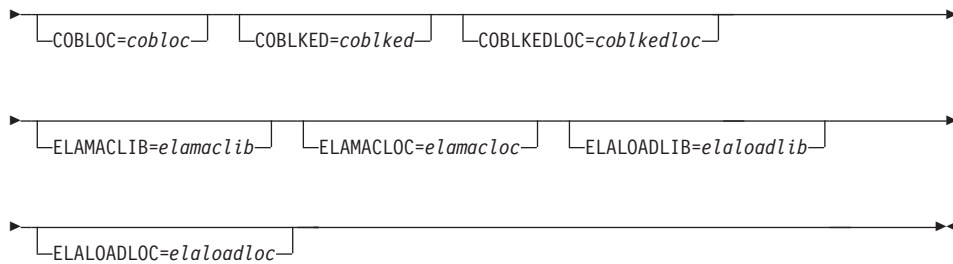
The ELAPREP EXEC prepares the VisualAge Generator parts for execution in the VM CMS or VM batch environments. Based on the VisualAge Generator part type, the ELAPREP EXEC does the following:

- SQL-prepare COBOL programs (generated VisualAge Generator programs)
- Compile COBOL programs (generated VisualAge Generator programs, mapping services programs, and tables)
- Link the compiled COBOL programs and generated object modules (map group format modules)

The ELAPREP EXEC is called from the VisualAge Generator program preparation REXX exec (program EXECP, which is also known as *appl.RXP* on the workstation) that is created by the VisualAge Generator.

Format





Parameters

MBR=

Indicates the VisualAge Generator part name being processed.

ENV=

Indicates the target environment; possible values are 'VMCMS' and 'VMBATCH'

COBOLTYPE=

Indicates the type of COBOL source.

DBCS=

This indicates whether Double Byte Character Set should be included in the compile. During the VisualAge Generator generation process, it is set to 'Y' if any DBCS characters are found within the VisualAge Generator program. Otherwise, it is set to 'N'.

PARTTYPE=

This indicates the file type that is being processed by this exec. Possible values that this exec processes are:

EZESRC -

COBOL source file

EZEFMT -

Map group format module

SQL=

This indicates whether the COBOL program contains SQL statements. During the program generation process, it is set to 'Y' if any SQL statements are found within the program. Otherwise, it is set to 'N'. If SQL=Y, then ELASQLAA EXEC is called to preprocess the COBOL program.

Defaults

If a variable refers to a <location>, it may be specified in any of these ways:

Variable	Location
ccccccc	VMLINK nickname
userid/vdev	Minidisk location
filepool:user.subdir	SFS directory

The ELAPREP EXEC parses the <location> string and issues a CMS VMLINK command.

VMFMODE=

CMS file mode where the object that is being processed resides; it must be a single alphabetic character. It is also the file mode of the LOADLIB in which the created module is stored. The file mode must be accessed in R/W mode. The default file mode is 'A'.

VMLOADLIB=

The target LOADLIB for the created part; the default is EZEAPPL.

COBLIST=

The listing destination for the COBOL compile; it must be PRINT, NOPRINT, or DISK. NOPRINT is the default and suppresses the generation of any listing. DISK generates a listing to the same file mode that contains the COBOL source. PRINT generates a listing to your virtual printer.

COBASM=

Is 'Y' for listing of assembler language expansion in the COBOL listing; possible values are 'Y' and 'N'. If 'Y', then the COBOL option LIST is used. If 'N', then the COBOL option OFFSET is used. The default is 'Y'.

COBOPT=

This indicates whether the OPTIMIZE option should be used for the COBOL compile. Possible values are 'Y' or 'N'. If 'Y', then the COBOL option OPTIMIZE is used. If 'N', then the COBOL option NOOPTIMIZE is used. The default is 'N'. NOOPTIMIZE can significantly increase the compile time performance, but OPTIMIZE might give faster runtime performance. Therefore, it might be beneficial to use the NOOPTIMIZE option during testing and use the OPTIMIZE option when moving the program to production.

COBTEST=

Indicates how the TEST option is used for the COBOL compile; possible values are 'Y' or 'N'. The default is 'N'. The following conditions apply:

- If 'Y', then the COBOL option TEST(ALL,SYM) is used.

- If 'N', then the COBOL option TEST(NONE,SYM) is used. The SYM suboption requests that symbolic variables be included in a formatted dump.

ERASETEXT=

Indicates whether ELAPREP should erase the text deck that was created by the COBOL compilation after using it; possible values are 'Y' or 'N'. The default is 'Y', which causes the text deck to be erased.

DATA=

Indicates whether DATA(24) or DATA(31) should be included in COBOL compilation; possible values are '24' and '31'. The default is '31'. This option controls whether data areas are acquired above or below the 16MB boundary (24 or 31 bit addressing).

AMODE=

Specifies the AMODE value that is used on the LKED step. Possible values are '24', '31', and 'ANY'. The default is '31'.

RMODE=

Specifies the RMODE value that is used on the LKED step. Possible values are '24' and 'ANY'. The default is 'ANY'.

SQLDBNAM=

This is the name of the SQL/DS database that is utilized during the preparation process. If SQLSTMDE=SINGLEUSER, then the default is SQLDBA. If SQLSTMDE=MULTIUSER, then the default is the last database used.

SQLPKGNM=

The name of the SQL/DS package name; it can be specified as fully qualified with the owner name. If you do not specify the name fully qualified, the default value for the owner name is the user ID specified in the SQLUSRPW symbolic parameter. The default value is the current program name. The maximum length for this parameter value is 17.

SQLPROPT=

The file name of the file that contains the SQL/DS VM precompiler options; it must have a file type of PREPPP. The first file found with the specified file name and a file type of PREPPP is used. The default value for VM is ELASQLPO. The maximum length for this parameter is 8.

SQLSTMDE=

The SQL/DS startup mode that is used by the preparation process. Valid values are MULTIUSER and SINGLEUSER. The default value is MULTIUSER.

SQLSTOPT=

The file name of the SQL/DS file containing the startup options that are used when starting SQL/DS in SINGLEUSER mode; it must have a file

type of SQLPARM. The first file found with the specified file name and a file type of SQLPARM is used. The default value is ELASQLSO.

SQLUSRPW=

The SQL/DS VM user ID and password that is utilized by the preparation process; the format is *'userid/password'*. The default value for SQL/DS is the VM user ID. The maximum length is 8 characters for the user ID and 8 characters for the password. These are separated by a slash (/), which results in a total maximum length of 17.

SQLLOC=

The <location> of SQL/DS; if not specified, the default is obtained from the installation options file.

USERLKED=

The name of the user TXTLIBs to utilize during the link edit step. This is useful when your VisualAge Generator program needs to use static linkage for a non-VisualAge Generator object deck. More than one TXTLIB can be specified by separating the TXTLIB names with a '/'. The minidisks or SFS directories containing the TXTLIBs must be accessed before calling ELAPREP. By default, ELAPREP will not use any user TXTLIBs.

USEREXECL=

This is the name of an exec that is called before ELAPREP invokes the CMS LKED command to link-edit the program. The purpose of this exec is to issue CMS FILEDEF commands for additional text decks that are to be included in the module. No parameters are passed to the exec and ELAPREP will not do any return code checking. The default is for no exec to be called.

COBLIB=

The names of the standard COBOL MACLIBs; more than one MACLIB can be specified by separating the TXTLIB names with a '/'. If not specified, the default is obtained from the installation options file.

COBLOC=

The <location> of the COBOL compiler and MACLIBs; if not specified, the default is obtained from the installation options file.

COBLKED=

The names of the TXTLIBs to use during the link edit of the COBOL program; more than one TXTLIB can be specified by separating the TXTLIB names with a '/'. If not specified, the default is obtained from the installation options file.

COBLKEDLOC=

The <location> of the TXTLIB to use during the link edit of the COBOL program. If not specified, the default is obtained from the installation options file.

ELAMACLIB=

The VisualAge Generator product MACLIB that is used during the COBOL compile. If not specified, the default is obtained from the installation options file.

ELAMACLOC=

The <location> of the VisualAge Generator product MACLIBs; if not specified, the default is obtained from the installation options file.

ELALOADLIB=

The VisualAge Generator product LOADLIB that is used during the link edit. If not specified, the default is obtained from the installation options file.

ELALOADLOC=

The <location> of the VisualAge Generator product LOADLIBs; if not specified, the default is obtained from the installation options file.

ELARUN EXEC

Purpose

The ELARUN EXEC starts a VisualAge Generator program on VM. It accesses the necessary disks, issues the necessary FILEDEF or DLBL commands, and then invokes an OSRUN of the program.

Format

(1) (2)
▶▶ ELARUN *appl*,

APPL,
<i>vmloadlib</i> ,

sqlflag ▶▶

Notes:

1. Do not specify nulls or blank spaces between any of the required or optional parameters.
2. All parameters are positional with commas used as separators.

Parameters

appl

The program name

vmloadlib

The LOADLIB where the program lives; the default LOADLIB is APPL.

sqlflag

'Y' or 'N'; indicates if the program uses SQL or not.

ELASQLAA EXEC

Purpose

The ELASQLAA EXEC invokes the SQL preprocessor for VisualAge Generator COBOL programs.

Format

▶—ELASQLAA—MBR=*mbr*—DBCS=Y—SQL=N—Options ▶
 └─DBCS=N—┘ └─SQL=Y—┘

Options

┌—VMFMODE=A—┐
└—VMFMODE=*vmfmode*—┘

┌—SQLLOC=*sqlloc*—┐ ┌—SQLDBNAM=*sqldbnam*—┐

▶—SQLPKGNM=*sqlpkgnm*—┐ ┌—SQLPROPT=ELASQLPO—┐ ┌—SQLSTMDE=MULTIUSER—┐
 └─SQLPROPT=*sqlpropt*—┘ └─SQLSTMDE=SINGLEUSER—┘

┌—SQLSTOPT=ELASQLSO—┐
└—SQLSTOPT=*sqlstopt*—┘

┌—SQLUSRPW=*userid/pw*—┐

Parameters

MBR=

This indicates the VisualAge Generator part name being processed.

DBCS=

This indicates whether Double Byte Character Set should be included in the compile. During the VisualAge Generator generation process, it is set to 'Y' if characters are found within the VisualAge Generator program. Otherwise, it is set to 'N'.

SQL=

This indicates whether the COBOL program contains SQL statements. During the program generation process, it is set to 'Y' if any SQL statements are found within the program. Otherwise, it is set to 'N'.

Defaults

If a variable refers to a <location>, it may be specified in any of these ways:

Variable	Location
ccccccc	VMLINK nickname
userid/vdev	Minidisk location

Variable	Location
filepool:user.subdir	SFS directory

The ELASQLAA EXEC parses the <location> string and issues a CMS VMLINK command.

VMFMODE=

CMS file mode where the object that is being processed resides; it must be a single alphabetic character. It is also the file mode of the LOADLIB in which the created module is stored. The file mode must be accessed in R/W mode. The default file mode is 'A'.

SQLLOC=

The <location> of SQL/DS; if not specified, the default is obtained from the installation options file.

SQLDBNAM=

This is the name of the SQL/DS database that is utilized during the preparation process. If SQLSTMDE=SINGLEUSER, then the default is SQLDBA. If SQLSTMDE=MULTIUSER, then the default is the last database used.

SQLPKGNM=

The SQL/DS package name; it can be specified as fully qualified with the owner name. If you do not specify the name fully qualified, the default value for the owner name is the user ID specified in the SQLUSRPW symbolic parameter. The default value is the current program name. The maximum length for this parameter value is seventeen.

SQLPROPT=

The file name of the file that contains the SQL/DS VM precompiler options; it must have a file type of PREPPP. The first file found with the specified file name and a file type of PREPPP is used. The default value for VM is ELASQLPO. The maximum length for this parameter is 8.

SQLSTMDE=

The SQL/DS startup mode that is used by the preparation process; valid values are MULTIUSER and SINGLEUSER. The default value is MULTIUSER.

SQLSTOPT=

The file name of the SQL/DS file containing the startup options that are used when starting SQL/DS in SINGLEUSER mode; it must have a file type of SQLPARM. The first file found with the specified file name and a file type of SQLPARM is used. The default value is ELASQLSO.

SQLUSRPW=

The SQL/DS VM user ID and password that is utilized by the preparation process; the format is 'userid/password'. The default value for SQL/DS is

the VM user ID. The maximum length is eight characters for the user ID and eight characters for the password. These are separated by a slash (/), which results in a total maximum length of seventeen.

Part 12. Utilities for VM Systems

Chapter 55. Using the Message File	
Conversion Utility on VM Systems . . .	533
Converting a Message File to an External	
Source Format Table.	533
Sample Exec for the Message File	
Conversion Utility	535

Chapter 55. Using the Message File Conversion Utility on VM Systems

The message file conversion utility enables you to create an external source format message file from an existing message file. Once the message file has been converted to external source format, the file can be imported into VisualAge Generator Developer to create the message table with its table contents. You can transfer message tables using the external source format import and export. Refer to the online help system provided with VisualAge Generator Developer for more information about the import and export facility.

Converting a Message File to an External Source Format Table

Perform the following procedures to convert an existing message file into an external source format table:

1. Allocate file MSGFILE to an existing VisualAge Generator message file.
2. Allocate file MSGTBLE to a sequential file to receive the external source format. The file must have 80-byte fixed-length records.
For VM, a CMS FILEDEF command (not a CMS DLBL command) must be used to define this file.
3. Allocate file MSGOPTS to the file which contains the conversion parameter statement. This should be a sequential file containing 80-byte fixed-length records. Figure 85 on page 534 shows the syntax of the conversion parameter statement.
For VM, a CMS FILEDEF command (not a CMS DLBL command) must be used to define this file.
4. Execute the message utility, DZGMSGC. The program resides in the VisualAge Generator Server library, ELALIB, and is started with the CMS OSRUN command. The VisualAge Generator Server and COBOL runtime libraries must be made global prior to starting the message utility.

Once the external source format file is created using the message file conversion utility, download the file to the workstation with VisualAge Generator Developer and import the file. This will load the message table part into the library. You can then generate the programs that use this message table.

Figure 86 on page 536 shows a sample EXEC that can be used to run the conversion utility. This sample exec is shipped with VisualAge Generator

Server for MVS, VSE, and VM and is called ELAMSGC EXEC. It must be tailored before it is used. Refer to the block comments contained within the exec for tailoring information.

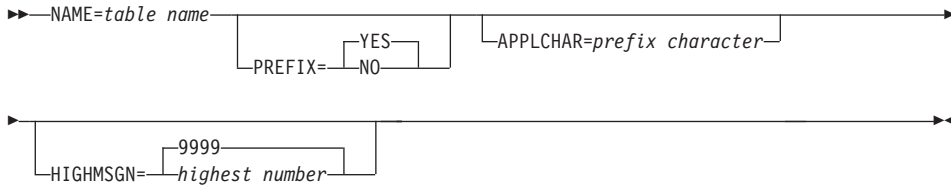


Figure 85. Message Table Conversion Parameters

NAME

Specifies the name of the table part to be created for VisualAge Generator Developer.

This keyword is required. The name can be 6 or 7 characters long, starting with an alphabetic character (A-Z). The remaining characters can be any combination of alphanumeric characters (A-Z, 0-9). The format of the name is `vvvvccc`, where `vvvv` is the value of the message table field specified in the program specifications, and `ccc` is a language code.

Valid language codes are as follows:

- ENU** U.S. English
- ENP** Uppercase English
- CHS** Simplified Chinese
- DEU** German
- DES** Swiss German
- ESP** Spanish
- JPN** Japanese (Kanji)
- KOR** Korean
- PTB** Brazilian Portuguese

PREFIX

Specifies whether the prefix should be included in the message text.

- YES** Specifies that the prefix should be included in the message text. The default is YES.
- NO** Specifies that the prefix should not be included in the message text.

APPLCHAR

Specifies the character that is to be inserted before the message number. If a character is not specified, the prefix is created with a blank before the message number. In message files, this character corresponds to the fourth character in the Message File field of the program specifications, or if there are less than 4 characters in this field, a blank.

HIGHMSGN

Specifies the highest message number to be converted. Valid values are 1 through 9999. The default is 9999. If the highest number is not known or you do not specify this parameter, the utility converts all messages in the message file.

The message conversion utility returns the following codes:

- 0** Message file conversion was successful
- 8** Error occurred during message conversion.

The return code is passed in register 15 through EZERCODE. This code can be checked for conditional execution of later steps in a job.

Sample Exec for the Message File Conversion Utility

Figure 86 on page 536 is a sample EXEC called ELAMSGC EXEC that can be used to run the conversion utility. It must be tailored before it is used. Refer to the block comments contained within the exec for tailoring information.

```

/*****/
/* ELAMSGC EXEC - Sample REXX exec to run the VisualAge */
/* Generator Server Message File Conversion */
/* Utility */
/* */
/* LICENSED MATERIALS - PROPERTY OF IBM */
/* 5648-B02 (C) COPYRIGHT IBM CORP. 1998 */
/* SEE COPYRIGHT INSTRUCTIONS */
/* */
/* STATUS: VERSION 1, RELEASE 2, LEVEL 0 */
/* */
/*****/
/* */
/* Several statements in this exec must be tailored to */
/* fit your configuration. They define variables that */
/* this exec will use to define: */
/* */
/* - The input VSAM dataset, the CSP Message file */
/* - The output ESF file (FIXED 80) */
/* - The file that contains the conversion */
/* parameter statement (FIXED 80) */
/* - The VSAM master catalog */
/* */
/* You will need to link and access the necessary */
/* minidisks for the above files before running this */
/* exec. */
/* */
/* Here is an example of the contents of a file */
/* containing a conversion parameter statement: */
/* 0 * * * Top of File * * * */
/* 1 NAME=TEST */
/* 2 PREFIX=yes */
/* 3 APPLCHAR=Y */
/* 4 HIGHMSGN=9999 */
/* 5 * * * End of File * * * */
/* */
/* You will need to link and access the necessary */
/* minidisks for the above files before running */
/* this exec. */
/* */
/*****/

```

Figure 86. Sample EXEC for Message Conversion Utility (Part 1 of 3)

```

/*****/
/* >> Tailor the lines below << */
/* >> Also, tailor the master catalog */
/* >> definition line later in this file! */
/*****/
esfout = 'MSGCONV ESFOUT C' /* Output ESF file */
vsamfilename = 'CSPMSG.DATA' /* Input VSAM dataset name */
vsamfilemode = 'F' /* Input VSAM dataset filemode */
msgoptsfile = 'MSGOPTS DATA C' /* File with msg conv stmt */

/*****/
/* Start Processing */
/*****/
ADDRESS COMMAND

'EXEC ELASETUP ALLOCATE'

EXEC_RC = FILE_LIST('ALLOCATE')

IF EXEC_RC = 0 THEN DO
  'EXEC ELARUN DZGMSGC,ELALIB,N'
  EXEC_RC = RC
  SAY 'MESSAGE CONVERSION COMPLETED WITH RETURN CODE=' EXEC_RC
END

TEMPRC = FILE_LIST('FREE')

'EXEC ELACLEAN FREE'

EXIT EXEC_RC

/*****/
/* FILE_LIST PROCEDURE: Call ELACALLC to allocate and free */
/* the necessary files. */
/*****/
FILE_LIST: PROCEDURE ,
  EXPOSE esfout vsamfilename vsamfilemode msgoptsfile
ARG ACTION

/*****/
/* Define the output ESF file. */
/*****/
ovrfm = ' ' /* override filemode */
'EXEC ELACALLC 'action',QSAM,WRITE,OLD,MSGTBLE',
  |', 'esfout', 'ovrfm,
  |', 'FB,80,80'
IF RC = 0 THEN DO
  SAY 'RETURN CODE' rc 'FROM ELACALLC 'action' FOR 'esfout
  RETURN RC
END

```

Figure 86. Sample EXEC for Message Conversion Utility (Part 2 of 3)

```

/*****/
/*                                          */
/* >> Tailor the statement below <<      */
/*   This DLBL command defines the master VSAM catalog.      */
/*   Refer to the CMS DLBL help file for information.        */
/*                                          */
/*****/
'DLBL IJSYSCT 'vsamfilemode' DSN MASTCAT ( VSAM PERM )'
IF RC ^= 0 THEN DO
  SAY 'RETURN CODE' rc 'FROM DLBL COMMAND FOR MASTER CATALOG'
  RETURN RC
END

/*****/
/* Define the input CSP Message File.      */
/*                                          */
/*****/
ovrfm = vsamfilemode /* override filemode */
'EXEC ELACALLC 'action',VSAM,READ,,MSGFILE',
  ||', 'vsamfilename', 'ovrfm
IF RC ^= 0 THEN DO
  SAY 'RETURN CODE' rc 'FROM ELACALLC 'action' FOR 'vsamfilename
  RETURN RC
END

/*****/
/* Define the input conversion parameter file.      */
/*                                          */
/*****/
ovrfm = '' /* override filemode */
'EXEC ELACALLC 'action',QSAM,READ,,MSGOPTS',
  ||', 'msgoptsfile', 'ovrfm
IF RC ^= 0 THEN DO
  SAY 'RETURN CODE' rc 'FROM ELACALLC 'action' FOR 'msgoptsfile
  RETURN RC
END

RETURN RC

```

Figure 86. Sample EXEC for Message Conversion Utility (Part 3 of 3)

Part 13. Diagnosing Problems on VM Systems

Chapter 56. Diagnosing Problems for VisualAge Generator Server on VM Systems

Systems	541
Detecting Errors	541
Reporting Errors	546
Controlling Error Reporting in VM CMS and VM Batch Environments	546
Error Reporting Summary	546
Using the VisualAge Generator Server Error Panel	547
Analyzing Errors Detected while Running a Program	548
Finding Information in Dumps	549
VisualAge Generator Server ABEND Dumps	549
COBOL or Subsystem ABEND Dumps	550
Information in the VisualAge Generator Server Control Block	551
Information in a Program, Print Services, or Table Profile Block	551
How to Find the Current Position in a Program at Time of Error	552
VisualAge Generator Server Trace Facility	552
Enabling VisualAge Generator Program Source-Level Tracing with a Generation Option	553
Activating a Trace	553
Activating a Trace Session for VM CMS or VM Batch	553
Deactivating a Trace Session	557
Printing Trace Output	557
Printing the Trace Output in VM CMS or VM Batch	557
Reporting Problems for VisualAge Generator Server	557
Problem Reporting Sequence	559

Chapter 57. Common Messages during Preparation for VM Systems

COBOL Compilation Messages	561
DB2/VSE Precompiler Messages	562

Chapter 58. Common System Return Codes for VM Systems

Common EZERT8 Return Codes	565
System Return Code Formats for EZERT8	565
Common System Return Codes in EZERT8	566
VisualAge Generator Server Return Codes	567
Common SQL Return Codes	573
Common VSAM Status Codes	575
OPEN request type	575
CLOSE request type	576
GET/PUT/POINT/ERASE/CHECK/ENDREQ request types	576
COBOL Status Key Values	576

Chapter 59. VisualAge Generator Server Return Codes and Abend Codes for VM Systems

Return Codes	579
ABEND Codes	579

Chapter 60. Codes from Other Products for VM Systems

Common System Abend Codes for All Environments	581
COBOL Runtime Messages	582
COBOL/370 Runtime Messages	583
Common COBOL Abend Codes	584

Chapter 56. Diagnosing Problems for VisualAge Generator Server on VM Systems

This chapter contains diagnosis, modification, or tuning information. Use this information to determine the source of the problem you encountered. Some common program definition, database, and system errors that might cause problems are described. This chapter also explains how to obtain error listings and diagnose runtime errors.

Detecting Errors

You can find most logic errors by using the test facility before you generate your program.

During generation, the preprocessor checks your program for any remaining syntax errors. In addition, VisualAge Generator also checks that your use of language elements is consistent with the resource association information you select for each file. The VisualAge Generator Developer also detects a few additional errors.

When you run your generated program, different types of errors are detected by VisualAge Generator Server, COBOL, or VM. The error handling varies depending on which product detects the error. Table 72 shows the types of errors that can occur when you run your generated program, as well as VisualAge Generator Server functions you can use for error detection and reporting.

Table 72. Types of Errors

VisualAge Generator Set Controls	Error Detected By	Error Handling
Screen Variable Edit Errors:		
The end user entered data that is not valid. For example:		
<ul style="list-style-type: none">• Alphabetic data in numeric field• Value not in allowable range• Omitted a required field		

Table 72. Types of Errors (continued)

VisualAge Generator Set Controls	Error Detected By	Error Handling
Use map definition to specify variable field edits or to include EZEMSG on the map.	Program logic or VisualAge Generator Server.	The error message is displayed on the map (in EZEMSG) and the cursor is placed on the field in error. If EZEMSG is not provided on the map, the ELAM01 screen is displayed with the EZEMSG error message.
File and Database Errors—Category 1:		
A hard error occurred in a file or database I/O operation. For example:		
<ul style="list-style-type: none"> • I/O error on the file or database • File not found 		
Use EZEFEFC to indicate whether the program provides logic to override normal database and file error processing for hard errors. Hard errors are defined in the online helps within VisualAge Generator Developer	Access method or database manager. The access method or database manager returns an error code to the generated logic or VisualAge Generator Server.	If EZEFEFC is set to 0 and a hard error occurs, the program ends with a transaction error, as described in Table 73 on page 547 . For SQL errors, the DB2/VSE messages describing the SQL error code are obtained from DB2/VSE. Refer to the DB2/VSE messages manual for your system for explanations of the error codes. If EZEFEFC is set to 1 and a hard error occurs, the program is responsible for controlling the error handling.
File and Database Errors—Category 2:		
A file or database error occurs due to incorrect program logic. For example, deleting or replacing a record without first doing an UPDATE process option is an example of this error type.		
Follow the file management rules in the VisualAge Generator help facility.	VisualAge Generator Server.	The program ends with a run unit error, as described in Table 73 on page 547.
File and Database Errors—Category 3:		

Table 72. Types of Errors (continued)

VisualAge Generator Set Controls	Error Detected By	Error Handling
Two generated programs which are being used together are defined in an inconsistent manner. For example, two programs have been generated with different file definitions.		
Follow the rules in the VisualAge Generator help facility.	COBOL or VisualAge Generator Server.	<p data-bbox="954 327 1243 474">If detected by VisualAge Generator Server, the program ends with a run unit error, as described in Table 73 on page 547.</p> <p data-bbox="954 495 1243 642">If detected by COBOL and either STAE is specified or you are using LE, a formatted dump is generated.</p> <p data-bbox="954 663 1243 751">If detected by COBOL and NOSTAE is specified, a system dump is generated.</p>
Arithmetic Overflow Errors:		
One of three situations occurs:		
<ul style="list-style-type: none"> <li data-bbox="315 852 1219 909">• Maximum value overflow when the result of an arithmetic operation does not fit into the maximum size number (18 digits) <li data-bbox="315 919 1219 976">• User variable overflow when the result of an arithmetic operation or a move to a numeric data item cause significant value (not decimal positions) to be lost <li data-bbox="315 987 483 1016">• Division by 0 		

Table 72. Types of Errors (continued)

VisualAge Generator Set Controls	Error Detected By	Error Handling
<p>Use the /NUMOVFL generation option to indicate that the generated program should check for overflow.</p> <p>If /NUMOVFL is specified, use the EZEOPER special function word to indicate whether program processing should continue if an overflow occurs.</p>	<p>If /NUMOVFL is specified, special logic is generated to intercept numeric overflow conditions. The EZEOPER special function word can be used to indicate the program provides logic to handle the overflow.</p> <p>If /NONUMOVFL is specified, overflow errors are not detected. The contents of the EZEOPER special function word are ignored. Truncation occurs on maximum value overflow, or user variable overflow. An abend occurs on division by 0.</p>	<p>If /NUMOVFL is specified and the EZEOPER special function word is set to 0 or 1, the maximum value overflow causes the program to end abnormally with a transaction error, as described in Table 73 on page 547 .</p> <p>If /NUMOVFL is specified and the EZEOPER special function word is set to 1, the user variable overflow also causes the program to end abnormally with a transaction error.</p> <p>In other situations, if /NUMOVFL is specified, the program continues to run.</p> <p>If /NONUMOVFL is specified, overflow errors are not detected. Division by 0 causes an abend 0C7. You need to code the program to ensure that overflow does not occur.</p>

Logic Errors:

The program ends because of faulty logic. For example:

- Numeric data that is not initialized
- Problems with array subscripting
- Infinite loops
- Moving incompatible data

Table 72. Types of Errors (continued)

VisualAge Generator Set Controls	Error Detected By	Error Handling
<p>Specifying the /INITRECD and /INITADDWS options force numeric data in records to be initialized during program initialization. However, it is still possible that numeric data in files or databases contains non-numeric values.</p>	<p>The test facility can be used to detect most of these errors before generation.</p> <p>If the COBOL STAE runtime option or the LE TRAP(ON) runtime option is specified, COBOL traps most of these errors.</p> <p>Otherwise, the VM operating system detects the errors.</p>	<ul style="list-style-type: none"> • If STAE is specified, or you are using LE, a formatted dump is generated. • If NOSTAE is specified, a system dump is generated.
Program Interaction Errors:		
<p>Two generated programs which are being used together are defined or run in an inconsistent manner; or, all parts of a generated program were not installed correctly. For example:</p>		
<ul style="list-style-type: none"> • A called program has been passed too few parameters. • A program with a First Map specification is being transferred to using a DXFR statement. • A transferred to program receives a map but does not have the First Map specification. • The program linkage table did not specify the same linkage conventions when the calling and called programs were generated. 		
<p>Follow the rules for the XFER and DXFR statements in the VisualAge Generator help facility.</p>	<p>VisualAge Generator Server.</p>	<p>The program ends with a run unit error, as described in Table 73 on page 547.</p>
<p>For VM CMS or VM batch, follow the guidelines in Chapter 52. Running a Generated Program on VM Systems. If special linkage conventions are used, ensure that the same linkage table is used for generating the calling and called programs.</p>		
System Errors:		

Table 72. Types of Errors (continued)

VisualAge Generator Set Controls	Error Detected By	Error Handling
The program was not properly installed. The following are common mistakes:		
<ul style="list-style-type: none"> • Missing ACB • VM CMS EXEC errors • Module cannot be found • Storage cannot be obtained 		
For VM CMS and VM batch, follow the guidelines in “Chapter 52. Running a Generated Program on VM Systems” on page 511.	VM operating system.	The program ends abnormally with a system abend in VM CMS or VM batch systems. For VM systems, a system abend dump is produced.

Reporting Errors

VisualAge Generator Server provides functions that help you determine the cause of a runtime problem. All runtime errors that VisualAge Generator Server traps are accompanied by error messages and supporting information to help diagnose the problem. Table 73 on page 547 shows the error diagnostic actions that can be taken based on the severity of the error and type of system being used.

Controlling Error Reporting in VM CMS and VM Batch Environments

The actions controlled by the runtime REXX exec are as follows:

Print message:

Messages are printed in the ELAPRINT data set.

Snap dump:

If a message indicates a snap dump was taken, the snap dump is produced in the ELASNAP data set.

Abend:

An abend dump is produced to the SYSABOUT data set.

Error Reporting Summary

The following table summarizes the error processing actions for VisualAge Generator Server.

Table 73. Error Processing Actions For VisualAge Generator Server Detected Errors

Category of Error	Action
Rununit error - The error is likely to occur for every user.	<p>VM CMS:</p> <ul style="list-style-type: none"> • Print message (ELAPRINT FILEDEF) • Snap dump determined by the message (ELASNAP FILEDEF) • Display error messages on terminal, if possible • Issue a rollback request if SQL databases were used • Set return code to 693 • Return <p>VM batch:</p> <ul style="list-style-type: none"> • Print message (ELAPRINT FILEDEF) • Snap dump determined by the message (ELASNAP FILEDEF) • Issue a rollback request if SQL databases were used • Set return code to 693 • Return
Catastrophic error - storage is corrupted or standard error reporting processing ends abnormally.	<p>VM CMS:</p> <ul style="list-style-type: none"> • Print message (ELAPRINT FILEDEF), if possible • Display error messages on terminal, if possible • Abend 16xx <p>VM batch:</p> <ul style="list-style-type: none"> • Print message (ELAPRINT FILEDEF), if possible • Abend 16xx
VisualAge Generator Server error at a point where standard error reporting process is not active	<p>All environments</p> <ul style="list-style-type: none"> • Abend, ABEND code indicates the reason for the error

See Table 74 on page 550 for information concerning the contents of the registers when a 16xx abend occurs.

Using the VisualAge Generator Server Error Panel

When an error occurs, VisualAge Generator Server attempts to display error messages on the current terminal. The panels used in displaying error messages are defined as map group ELAxxx where xxx is the language code. External source format for the map group can be found on the installation user ID's local sample disk as ELAxxx ESFSRC, so that you can customize the constant information on your map to your own installation. For example, you

can add the phone number of your system administrator so that the user encountering the error knows who to call to report the problem.

Refer to the *Program Directory for VisualAge Generator Server for VM* for information about tailoring the panel.

The following figure shows the error panel (map ELAM02 in the map group) as it is shipped with the product. The panel shows the same diagnostic information that is written to the ELAPRINT file. If there are more error messages than can fit on a single panel, the last line on the panel prompts the user to press a key to display additional error messages.

```
Unexpected Program Failure

An unexpected input/output or program error occurred in the
program you were running. Please make a note of the program
name, date, time, and initial error messages and report them to your
system administrator.

Program name ... A8CCCSP
Date ..... 08/01/96
Time ..... 08:26:50

Error Messages:

ELA00093I An error occurred in program A8CCCSP, process or group PDFI
ELA00073P SQL error, command = PREPARE, SQL code = -551
ELA00074I SQL error message: RANDOPM ,RUN/EXECUTE,TINK.A8CCCSP

PF3=EXIT
```

Figure 87. Panel ELAM02 (example).

Analyzing Errors Detected while Running a Program

Use the error messages, diagnostic messages, and the information in Chapter 61. VisualAge Generator Server Messages for MVS, VSE, and VM Systems to determine the cause of the problem. If the error is detected by another product (for example, COBOL), check the information in Chapter 58. Common System Return Codes for VM Systems and Chapter 60. Codes from Other Products for VM Systems and the documentation for the other product. If you cannot determine the cause of the problem using this information and if the problem can be created again in the test environment, use the test facility to isolate and debug the error before generating your program.

For more information about testing with the VisualAge Generator Developer interactive test facility, refer to the VisualAge Generator help facility. Refer to the COBOL manuals for your versions of these products for additional information on their debugging facilities.

If you get an error for the runtime REXX exec, check the generation listing for the programs involved for any error messages related to exec generation. In addition, ensure the tailoring that was done on the runtime templates is correct. Also check the changes you made when you customized the sample runtime REXX exec.

When abends occur, the problem determination might require assistance from the IBM Support Center. In this case, be prepared to provide IBM with the following information:

- VisualAge Generator Developer print of the problem program
- COBOL compilation listing created using the COBLIST=DISK user-defined symbolic parameter option
- Formatted dump
- VisualAge Generator Server diagnostic information written to ELAPRINT data set

IBM requests a COBOL debugger trace listing only if the information is needed for problem determination. IBM will give you the information on how to specify the trace options if the information is necessary.

Finding Information in Dumps

Information about the problem program can be determined by finding the address of the VisualAge Generator Server control block in a dump.

VisualAge Generator Server ABEND Dumps

If the dump code is 1600, 1602, or ELAE, the dump was initiated because VisualAge Generator Server detected an error. Register 2 at ABEND points to the VisualAge Generator Server control block. Register 4 points to a linked list of messages formatted as shown in Figure 88 on page 550.

Table 74. Registers when a SNAP dump is taken or a VisualAge Generator Server ABEND occurs.

Reg.	Value
2	Points to VisualAge Generator Server control block. At offset 272 (hexadecimal offset 110) from the start of the VisualAge Generator Server control block is the address of the initial program profile block, which provides information about the first generated program that was started. At offset 276 (hexadecimal offset 114) from the start of the VisualAge Generator Server control block is the address of the current program profile block, which provides information about the VisualAge Generator program that was running at the time of the abend.
4	Points to the message buffer that contains all messages.

The following diagram shows the format of the message buffer that contains all the messages in the dump.

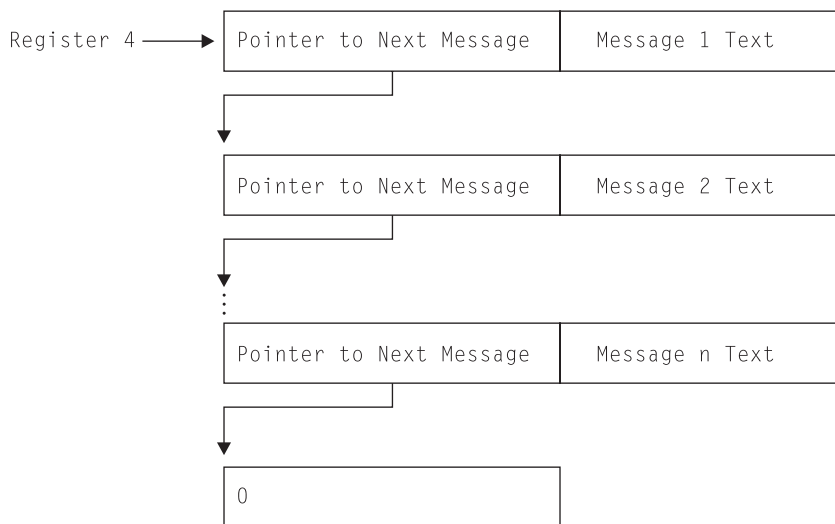


Figure 88. Message Buffer Format

COBOL or Subsystem ABEND Dumps

If the dump is not a VisualAge Generator Server abend, locate the VisualAge Generator Server control block by finding the string ELARHAPP followed immediately by a program name. ELARHAPP is the identifier at the start of a program profile block. The four-byte address at hex offset 20 in the program profile is the VisualAge Generator Server control block address. If 0, the program might not yet be activated. Do a search for another ELARHAPP control block followed by a program name. See Table 76 on page 551 for information in the program profile block.

Information in the VisualAge Generator Server Control Block

The following information is in the VisualAge Generator Server control block:

Table 75. Information in the VisualAge Generator Server Control Block

Offset in hex	Length in bytes	Contents
0	16	Control block identifier - *EZERTS-CONTROL*
110	4	Program profile address for current program
114	4	Program profile address for initial program
120	8	User identifier
1CC	18	Current process or group

Information in a Program, Print Services, or Table Profile Block

Each generated COBOL program contains a profile control block in COBOL working storage initialized with information about the program. The first eight bytes contain an eye-catcher constant identifying whether the program is an application, print services program, or table program. The second eight bytes contain the program name. Other information in the profile block is shown in the following table:

Table 76. Locator Format for Generated COBOL Program Dumps

Offset in hex	Length in hex	Contents
00	08	Program type identifier: <ul style="list-style-type: none">• ELARHAPP — Application program• ELAAHMGC — Print services program• ELARMTTP — Table program
08	08	Program name
10	08	Program generation date (MM/DD/YY)
18	08	Program generation time (HH:MM:SS)
20	04	VisualAge Generator Server control block address
24	02	Generator version
26	02	Generator release
28	02	Generator modification level
2A	10	Reserved
34	08	Target runtime system

How to Find the Current Position in a Program at Time of Error

The VisualAge Generator Server control block identifies the currently running program and process or group at the time of the error (Table 75 on page 551). Associated error messages identify the VisualAge Generator statement number for errors detected by VisualAge Generator Server that need statement identification to resolve the problem. For performance reasons, the generated COBOL program does not keep track of the VisualAge Generator statement number for each generated statement. If a program exception occurs in a generated program, you can determine the VisualAge Generator statement number by finding the COBOL statement that was not successful in a COBOL program listing that contains the VisualAge Generator statements generated as comments.

VisualAge Generator Server Trace Facility

The VisualAge Generator Server trace facility can be used by the IBM Support Center to aid in problem determination, or by the program user to trace program activity.

There are two levels of tracing available:

- VisualAge Generator program source-level tracing
- VisualAge Generator Server runtime-level tracing

With source-level tracing, you can request traces of VisualAge Generator statements, traces of the data, and error codes after every SQL call in a program, except SQL calls made with the SQLEXEC process option. Source-level tracing is enabled with the use of the /TRACE generation option. Source-level tracing is activated in the runtime environment by specifying trace filter criteria. See “Activating a Trace” on page 553 for more information on activating traces.

With runtime-level tracing, you can request a data stream trace, a VisualAge Generator Server internal dump trace, or a service routine trace. Runtime-level tracing does not require the use of a generation option. Runtime-level tracing is activated in runtime environment by specifying trace filter criteria. See “Activating a Trace” on page 553 for more information on activating traces.

Use these functions only with the assistance of the IBM Support Center. If you use these functions without assistance, large amounts of trace output might be produced based on trace option selection.

Enabling VisualAge Generator Program Source-Level Tracing with a Generation Option

You must specify the /TRACE generation option in order to get source-level trace output.

The VisualAge Generator Developer preprocessor validates the generation option and its parameters. VisualAge Generator Developer creates the necessary COBOL code to accomplish the type of tracing that you request. The /TRACE generation option can be specified with these parameters:

NONE	VisualAge Generator program source-level tracing is not enabled NONE is the default option. /TRACE=NONE excludes any other /TRACE generation options.
SQLERR	Enables tracing of only the SQL error codes returned in the SQL Communications Area (SQLCA) after an SQL call
SQLIO	Enables tracing of both the data and the SQL error codes returned in the SQL Communications Area (SQLCA) after an SQL call Once a program has been generated with the /TRACE=SQLIO generation option, you can then turn on the SQLIO trace option, the SQLERR trace option, or both, or neither.
STMT	Enables tracing of each VisualAge Generator statement that is encountered during the running of the program

Note: More than one parameter can be specified in the /TRACE generation option. The parameter must be separated by comma. The following example shows both SQLIO and STMT are traced:

```
/TRACE=SQLIO,STMT
```

Refer to the *VisualAge Generator Generation Guide* for more information on generation options.

Activating a Trace

Tracing is activated during run time by specifying the ELATRACE DD name in the runtime exec for the VM CMS or VM batch environments.

Activating a Trace Session for VM CMS or VM Batch

Tracing is activated by providing trace filters in a preallocated data set with the DD name ELATRACE before starting the program or job to be traced. A

FILEDEF must be specified for ELATRACE and another FILEDEF must be specified for ELATOUT. The FILEDEF for ELATOUT defines where the trace output is to go.

ELATRACE contains control statements which control the programs and events to be traced. The attributes for the data set are LRECL=80, DSORG=PS, and RECFM=FB. If the ELATRACE data set is empty or allocated as DD DUMMY, all services are traced, data streams are not traced, and SQL I/O, SQL errors, and program statement are not traced even if enabled through a /TRACE generation option. The attributes for the ELATOUT data set are LRECL=133 and RECFM=F. ELATOUT can be directed to a virtual printer or to a data set. Figure 89 on page 555 shows the correct syntax for the trace control statements.

The following attributes can be used with the :FILTER statement:

- DATASTREAM=Y | N
If DATASTREAM=Y is specified, the 3270 data streams built or received by VisualAge Generator Server are traced. The default value is N, no data stream tracing.
- TRACETOFILE=Y | N
If TRACETOFILE=Y is specified, the trace output is directed to the preallocated data set named ELATOUT in addition to being directed to an in-storage wrap-around trace buffer.
If TRACETOFILE=N is specified, the trace output goes to an in-storage wrap-around trace buffer. The size of this trace buffer is defined during customization of VisualAge Generator Server. The default for the TRACETOFILE option is N.
TRACETOFILE=Y must be specified if SQLIO=Y or SQLERR=Y are specified. All trace output for SQL I/O and SQL errors is directed to the ELATOUT data set, not to the in-storage wrap-around trace buffer.
- APPSTMT=Y | N
If APPSTMT=Y is specified, the trace facility provides the process name and the statement for each VisualAge Generator statement that the program processes. You must use the /TRACE=STMT generation option to enable this type of tracing. The default for the APPSTMT option is N.
- SQLIO=Y | N
If SQLIO=Y is specified, the trace facility provides traces of the data and error codes on the return from the SQL call. The VisualAge Generator SQL process name, the process option, the process object, the SQL function name, and the VisualAge Generator data item name, length, type, and contents are given. You must use the /TRACE=SQLIO generation option to enable this type of tracing. The default for the SQLIO option is N.
- SQLERR=Y | N
If SQLERR=Y is specified, the trace facility provides traces of the error information that comes back from SQL on every database call. The SQLCODE, SQLERRP, SQLSTATE, SQLWARN, SQLERRD, SQLEXT, and SQLERRMC codes are given. You must use the /TRACE=SQLERR or the /TRACE=SQLIO generation option to enable this type of tracing. The default for the SQLERR option is N.
- IDUMP=Y | N
If IDUMP=Y is specified, the trace facility provides dumps of certain VisualAge Generator Server internal storage areas. The default for the IDUMP option is N, no internal storage dumps.

:APPLS

Application program names or print service program names

If program names are specified, only the specified programs are traced. Otherwise service requests from each generated program are traced. Up to 16 program names can be specified.

:SERVICES

Service numbers

If service numbers are specified, only those specific services are traced. To trace all service numbers, 999 must be specified. Otherwise, up to 32 service numbers can be specified.

Note: The entry to ELARSINI (initialization service) and the exit from ELARSTRM (cleanup service) are not traced. ELARSINI initializes the trace facility. ELARSTRM ends the trace facility.

Deactivating a Trace Session

To stop tracing in VM CMS or VM batch, cancel the program and remove the ELATRACE and ELATOUT DD cards from the runtime REXX exec.

Printing Trace Output

If the ELATOUT data set is not allocated for VM CMS or VM batch, trace output is written to a wrap-around trace buffer in memory. The trace output can be seen in dumps taken when programs end abnormally.

Printing the Trace Output in VM CMS or VM Batch

Trace output is directed to the ELATOUT DD statement and is printed as directed on the DD statement.

Reporting Problems for VisualAge Generator Server

VisualAge Generator provides a set of manuals that contain information that can assist you in using the VisualAge Generator products. Check these manuals first for answers to your problems or questions. A list of the VisualAge Generator manuals are located in the Preface section of each manual.

Documenting information is vital not only in locating the source of the problem, but also in resolving the problem.

Gather the necessary documented information before you contact IBM Software Support to report your problem. The following list outlines the minimum required documentation.

- The release level of VisualAge Generator Server and the operating system it is running under
- The list of all PTFs and additional APARs that have been applied to VisualAge Generator Server
- All VisualAge Generator and non-VisualAge Generator error messages including text and return codes
- The first indication that the problem has occurred
- A detailed description of the problem and any additional information attributing to the problem
- Determine whether the problem occurred during generation, compilation, or run time
- Can the problem be recreated?
- Does the program run successfully under VisualAge Generator Developer?
- Do other VisualAge Generator generated programs fail? If not, does the problem program do something unique that the programs that work do not do?
- Does the program interface with non-VisualAge Generator programs? If so, what language are the non-VisualAge Generator programs written in?
- Any recent changes made to VisualAge Generator Server, the program, VM, DB2/VSE, or other related software
- The terminal types involved if the problem is terminal related
- Languages installed and the one used to create the problem again

Your IBM Support Center representative might request documentation depending on the type of problem. This documentation might include the following:

- The program with associates exported in external source format
- The generated COBOL source code
- The generation options
- Output from preparation or run time
- DB2/VSE information (SQL preparation results, table definitions, and so forth)
- Dump of the error

These files can be put on tape or cartridge and sent to the IBM Support Center. Please follow these instructions when creating the tape or cartridge:

1. Use ONLY IBM standard utilities, such as VMFPLC2, TAPE, or AMSERV.
2. Data written to cartridges may be compacted. Tell the IBM Support Center representative if the cartridge was recorded in compacted format.
3. Process any dumps using the DUMpload command.

4. Provide the commands used to create the tape.
5. Any data you supply should be marked with your:
 - Company name
 - Contact Name
 - Problem number

Problem Reporting Sequence

When you report a problem, inform your IBM Support Center representative of the symptom string you received if an ABEND occurred. A symptom string contains an ABEND code, the name of the VisualAge Generator Server module in control at the time of the error, and the program ID.

Figure 90 shows the sequence of events that should occur after you describe the problem to the IBM Support Center.

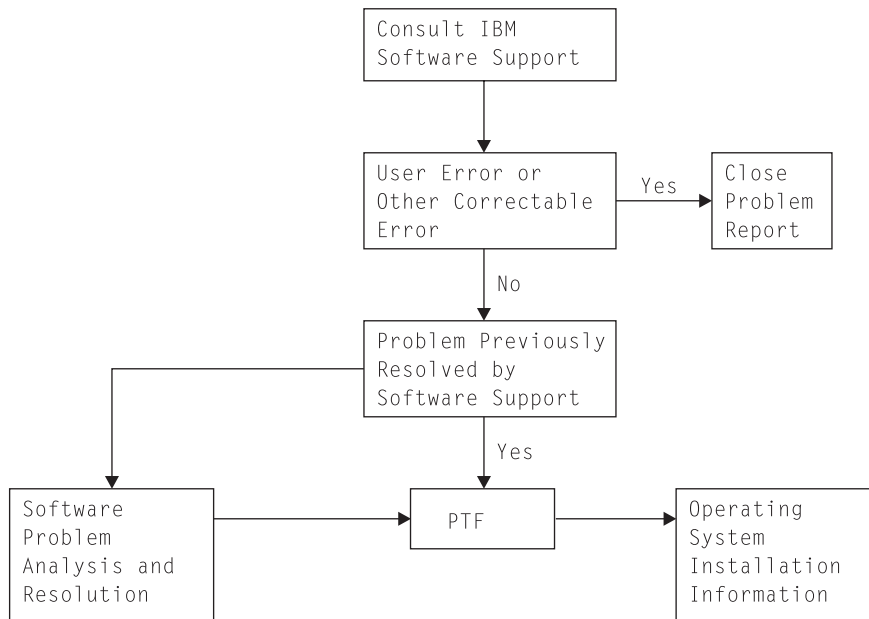


Figure 90. Problem Reporting Sequence

If the IBM Support Center has an existing solution for the problem, see the appropriate system documentation for instructions on applying the patch (if the resolution is a temporary patch) or program temporary fix (PTF).

If the IBM Support Center cannot resolve the problem, or a PTF is unavailable, an Authorized Program Analysis Report (APAR) number is assigned to your problem. You might be asked to assist in collecting information to submit an APAR.

Chapter 57. Common Messages during Preparation for VM Systems

This chapter contains some error messages from other products. It is not a complete list. For a complete explanation of product messages, refer to the documentation provided with that product.

COBOL Compilation Messages

Only the most frequently occurring COBOL compilation messages are listed in this section. If you receive other compilation messages that start with IGY or if you need a more complete explanation of one of the messages, refer to the documentation for the COBOL product and release you are using.

IGYPS2015I The paragraph or section prior to paragraph or section EZEMAIN-PROCESS did not contain any statements.

IGYPS2023I Paragraphs prior to section EZEMAIN-PROCESS were not contained in a section

Explanation: One or both of these messages occur if your program has been processed by the DB2/VSE precompiler.

User Response: They are normal messages that you can ignore.

IGYOP3091W
Code from “?” to “?” can never be executed, and was therefore discarded.

IGYOP3093W
The “PERFORM” statement at “?” cannot reach its exit.

IGYOP3094W
There may be a loop from the “PERFORM” statement at “?” to itself. “PERFORM” statement optimization was not attempted.

Explanation: These messages can occur separately or in combination with each other if your program has been processed using the OPTIMIZE compiler option.

User Response: These are normal messages that you can ignore.

IGYPA3013W

Data item “?” and “?” had overlapping storage. An overlapping move will occur at execution time.

Explanation: This message occurs if your program attempts to assign the value of a data item to the same data item.

User Response: You might want to check that you really intended to do this.

IGYPG3113W

Truncation of high-order digit positions may occur due to precision of intermediate results exceeding 30.

Explanation: This message might occur if your program was generated with the /MATH=COBOL generation option.

DB2/VSE Precompiler Messages

Only the most frequently occurring DB2/VSE precompiler messages are listed in this section. If you receive other messages that start with ARI or if you need a more complete explanation of one of the messages, refer to the documentation for your release of DB2/VSE.

ARI0501I AN SQL WARNING HAS OCCURRED. SQLCODE DESCRIPTIVE TEXT.

Explanation: This message is followed by message ARI505I, which provides the SQLCODE and ROWCOUNT information. The SQLCODE descriptive text describes the SQLCODE error condition. If SQLCODE is 204, one or more SQL tables have not been created. The tables that do not exist are identified in an explanation associated with the message by:

xxxxxxx was not found in the system catalog where xxxxxxx is the table name.

User Response: Create the necessary SQL tables and synonyms before the running the program.

ARI054E SQL STATEMENT CONTAINS MORE THAN 256 HOST VARIABLES.

Explanation: A limit of the host variable reference in a single SQL statement has been exceeded.

User Response: Change the SQL statement to contain fewer than 256 host variables.

ARI0565I THE INPUT SOURCE CONTAINS EITHER NO SQL STATEMENTS, OR NO SQL STATEMENTS THAT WILL CREATE A PACKAGE.

Explanation: This can occur if one of the following is true:

- The program contains no SQL process options.

- The program contains only CALL EZECONCT, EZEROLLB, RESET or EZECOMIT statements. These calls result in CONNECT, ROLLBACK or COMMIT SQL statements that do not create a package.

User Response: The preprocessing continues. The generated program can be run successfully. If you have SQL process options that result in SQL calls other than CONNECT, ROLLBACK, or COMMIT in your program, then you should not receive this message.

ARI5387E The SQL statement length exceeds 8192 positions

Explanation: A limit of the SQL statement length has been exceeded. The possible causes are the following:

- The SQL statement is too long.
- The end of statement delimiter is missing.
- A FOR UPDATE OF clause is added to the end of a static query statement.

- The expansion of a host structure reference has increased the size of the SQL statement.

User Response: Either break the SQL statement into smaller statements or remove unnecessary conditions that might exist in the statement.

ARI5400I Warning: statement-type is not supported.

Explanation: The SQL statement shown is not supported by the DB2/VSE system. The statement-type can be either DECLARE TABLE, DECLARE STATEMENT, or DISCONNECT.

User Response: If your program depends on the code expansion, change your program so that it does not depend on either of the above statements. Or, install the release of DB2/VSE that supports this statement-type.

Chapter 58. Common System Return Codes for VM Systems

The information within this chapter is diagnosis, modification, or tuning information.

VisualAge Generator Server messages might include return codes from COBOL, SQL, CMS, or VSAM. This chapter contains only the most common errors that occur during file input and output operations.

Common EZERT8 Return Codes

The EZERT8 special function word contains a return code indicating a reason that a file I/O process option or a CALL to a service routine is not successful. Codes specific to the system or the access method are returned when the user requests system codes with the /SYSCODES generation option.

If the /NOSYSCODES generation option is specified, the system return codes are converted to VisualAge Generator Server return codes. This allows programs developed previously to receive the same return codes as before.

System Return Code Formats for EZERT8

For the VM environment, if EZERT8 is in the form *RSnnnnnn*, look under *nnnnnn* in “Common System Return Codes in EZERT8” on page 566.

SEQ: EZERT8 contains the COBOL status key value or values in the first 2 characters. The remaining 6 characters are zeros.

SEQRS: The contents of EZERT8 depend on the operation that was not successful:

- If a dynamic allocation is not successful, EZERT8 contains the value '00000218'.
- If an OPEN is not successful, EZERT8 contains return code 8 ('00000008').
- If a READ end-of-file condition occurs, EZERT8 contains return code 4 ('00000004').
- If a READ, WRITE, or CLOSE is not successful, EZERT8 contains return code 12 ('00000012').

VSAM: EZERT8 contains the COBOL status key value or values in the first 2 characters followed by 2 characters for the COBOL VSAM return code (VSAM

feedback code), 1 character for the COBOL VSAM function code (VSAM component code), and 3 characters for the COBOL VSAM feedback code (VSAM reason code).

VSAMRS: The operation that is not successful determines the contents of EZERT8:

- If a dynamic allocation is not successful, EZERT8 contains the value '00000218'.
- If an OPEN or CLOSE is not successful, the first 2 bytes of EZERT8 contain the error code from the VSAM application control block (ACB) in hexadecimal. The remaining 6 characters are zeros.
- If an operation other than OPEN or CLOSE is not successful, the first 2 characters are zeros followed by 2 characters for the COBOL VSAM return code (VSAM feedback code), 1 character for the COBOL VSAM function code (VSAM component code), and 3 characters for the COBOL VSAM feedback code (VSAM reason code).

For VSAM codes, refer to the *VSE VSAM Macro Instruction Reference*.

Common System Return Codes in EZERT8

The following table gives an explanation of the most common values that you receive in EZERT8 when system codes are requested. If your return code is not listed here, or you would like more information, refer to the appropriate manuals for your environment.

Table 77. EZERT8 return codes for VM.

Return code	Meaning
A0000000	VSAM open error - empty indexed file
BC000000	VSAM open error - file is not in VSAM format
00000004 on non-VSAM file	End of file
00000008 on non-VSAM file	Error opening file
00000012 on non-VSAM file	Error on I/O or closing a file
0008*004 for nonrelative	End of file
0008*004 for relative	No record found
0008*008 for an ADD request	Duplicate record
0008*016 if scanning an indexed record	End of file
0008*016 if not scanning an indexed record	No record found
0008*028	File full
0008*116	No record found
*****74	No record found
	Note: * represents any character.
060F0000 on VSAM file	End of file
06810000 on VSAM file	No record found
06820000 on VSAM file	Duplicate record
06830000 on VSAM file	File is full

VisualAge Generator Server Return Codes

The return codes list is sequenced by return code, with the alphabetic return codes (A to Z) occurring before the numeric return codes (0 to 9).

Table 78. VisualAge Generator Server Return Codes

RC	Component	Probable Cause
<i>Cnn</i>	File control/request	These return codes do not have a VisualAge Generator mnemonic. VSAM returned a return code of 12 on file input/output. The <i>nn</i> is replaced by the VSAM reason code from the feedback field. For more information, see the <i>VSE VSAM Macro Instruction Reference</i> manual.
02	EZCHART	An incorrect number of parameters was passed on a call to EZCHART.
03	EZCHART	GETMAIN error - An attempt to obtain storage for data conversions was not successful.
04	EZCHART	A data overflow occurred during the conversion of numeric data to floating point format. Data must fit into 6-place floating point format.
06	EZCHART	The printer depth was not specified in Chart Control.
07	EZCHART	The printer width was not specified in Chart Control.
08	EZCHART	The number of print copies was not specified in Chart Control.
09	EZCHART	The printer name was not specified in Chart Control.
10	EZCHART	A load or link error occurred for a GDDM* routine.
52	Terminal support	You attempted to enter VisualAge Generator on an unsupported device (such as a 3278-52 terminal). This device is not supported by VisualAge Generator.
101	Message processing	The message was truncated.
102	Contents control	The module specified on a LOAD macro is already in storage.
102	File control/request	The end of file was reached. The VisualAge Generator mnemonic EOF is set. Note: EOF is set when a SCAN is performed on an empty file.

Table 78. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
103	File control/request	You performed an operation on a record that has a duplicate key, or a key in the record for an alternate index is duplicated. The I/O operation completed, and the VisualAge Generator mnemonic DUP is set.
104	File control/request	The end of file was reached. The VisualAge Generator mnemonic EOF is set.
20B	Storage allocation	Operands that are not valid were specified on either a GETMAIN or FREEMAIN macro.
20C	Defined data set	The data set name specified on an issued DEFDS command already exists in the external work file.
20C	Storage allocation	An error occurred while processing a FREEMAIN macro.
200	Service request	An service request was issued that is not valid. This is a system error. Contact the IBM Support Center.
201	File open/connect	The connection already exists. The possible cause is a file with the same name is already in use. VisualAge Generator mnemonic FNA is set.
201	Message processing	Variables were passed to be built into the message, but the message contained no variable fields; or, the message contained variable fields, and no variables were passed for them.
201- 206	Service request	Service request errors occurred while processing a DEFDS command. This is a system error. Contact the IBM Support Center.
202	File control/request	Record not found. The VisualAge Generator mnemonic NRF is set.
202	Storage allocation	The ORIGIN specified on a FREEMAIN macro does not match storage already in use.
203	File control/request	The record was not found. The VisualAge Generator mnemonic NRF is set.
203	Storage allocation	Either the ORIGIN specified on a FREEMAIN macro does not begin on a doubleword boundary, or 0 LENGTH was specified on a GETMAIN.
204	Storage allocation	An attempt has been made to free storage that has not been allocated or that has already been freed.

Table 78. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
205	File control/request	The record was not found. The VisualAge Generator mnemonic NRF is set. Note: The VisualAge Generator mnemonic NRF is set when a SCAN or SCANBACK is preceded by a SET record SCAN on an empty indexed file.
205	Storage allocation	The LENGTH specified on a FREEMAIN macro is 0.
206	File control/request	You attempted to store a record with a duplicate key while using an index that does not allow duplicate keys. The VisualAge Generator mnemonics UNQ and DUP are set.
207	File control/request	The record was not found. The VisualAge Generator mnemonic NRF is set.
209	Service request	The name specified by the NDSNAME on an ALTDS command already exists in the external work file. This is a system error. Contact the IBM Support Center.
210- 211	Service request	Miscellaneous errors occurred on an ALTDS request. This is a system error. Contact the IBM Support Center.
212	Service request	An I/O error occurred while copying data from the work file to the external work file during an ALTDS service request.
213	Service request	The COPIES operand on a SUBMIT.PRINT service request is not valid. This is a system error. Contact the IBM Support Center.
214	Service request	The data set on a SUBMIT.PRINT service request cannot be found. This is a system error. Contact the IBM Support Center.
215	File control/request	You attempted to store a record with a duplicate key while using an index that does not allow duplicate keys. The VisualAge Generator mnemonics UNQ and DUP are set.
216	File open/connect	A connection was attempted to an ESDS file or transient data in direct mode. The VisualAge Generator mnemonic FMT is set.
216	Service request	The data set specified on a DEFDS request matches a CONNECT already in use. This is a system error. Contact the IBM Support Center.

Table 78. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
217	File open/connect	An attempt was made to subconnect to a serial file. Check to see if a called program is attempting to reference the same serial file that has been referenced by the calling program.
217	Service request	<p>A PRINT error has occurred for one of the following reasons:</p> <ul style="list-style-type: none"> • An error occurred while writing to the transient data queue on MVS CICS. The most common errors are QIDERR, IOERR, LENGERR, and NOSPACE. • An error occurred while writing to the data definition name (DD name) EZEPRINT in VM or MVS batch. A possible cause is that the printer file (for example, EZEPRINT) has been allocated incorrectly or not been allocated at all. You get this error, for example, if the data set allocated for the print output has the wrong record format (anything other than VBA) or the wrong record length (shorter than the print output line length). You get this error if the allocation statement (in the CLIST, startup JCL, or runtime REXX exec) is missing or incorrect.
218.	Service request	The file is not available. The VisualAge Generator mnemonic FNA is set.
22A	File control/request	The available storage space has been exhausted. Try the operation again. If the problem persists, contact your system programmer.
220	File open/connect	<p>A format error occurred. Either the characteristics of a file are not supported by VisualAge Generator Developer, or they are incompatible with the VisualAge Generator Developer record. The VisualAge Generator mnemonic FMT is set.</p> <p>In MVS, this occurs if no member name was specified for gaining access to a PDS data set.</p>
220	File control/request	The record length for a file is larger than the maximum record length defined in the system.
223	Service request	The attach of the print subtask was not successful, or the print subtask abended. This is a system error. Contact the IBM Support Center.

Table 78. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
225	Service request	The print subtask abended. This is a system error. Contact IBM Software Support.
25A	File control/request	The data set cannot be extended because VSAM cannot allocate additional direct-access storage space. Either not enough space is left to make the secondary allocation request, or you attempted to increase the size of a data set while processing with SHROPT=4 and DISP=SHR. The VisualAge Generator mnemonic FUL is set.
251	File open/connect	For MVS or VM environments, either the DD name has not been allocated, or the data set for the dynamic allocation does not exist.
280	File control/request	The data set that you are trying to connect to is already in use. A probable cause is that your program has a data set associated to one record and you are trying to use another record with the same data set associated. You need to issue a CLOSE on the first record to free the data set before trying to use it with another record.
291- 294	Service request	A mapping error occurred.
291- 294	Terminal support	A mapping error occurred.
380	File control/request	A deadlock occurred. One transaction is attempting to update a record that is currently locked by another transaction. The VisualAge Generator mnemonic DED is set.
389	File control/request	The resource control block could not be found to process the request against. This is a system error. Contact the IBM Support Center.
4nn	File open/connect	For MVS or VM, the VSAM GENCB for an ACB was not successful. The value of <i>nn</i> is determined from VSAM return codes. If register 15 contains 4, <i>nn</i> is replaced by the contents of register 0. If register 15 does not contain 4 (or 0), <i>nn</i> is replaced by 50 plus the contents of register 15.
5nn	File open/connect	For MVS or VM, an OPEN request is not successful. For VSAM files, a SHOWCB for the ERROR field is done subsequent to the open error. The value of the ERROR field replaces <i>nn</i> . For non-VSAM sequential files (QSAM), <i>nn</i> is replaced with a value of 0. For spool files in a CICS environment, <i>nn</i> is also replaced with 0.

Table 78. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
5A0	File open/connect	An attempt was made to open a VSAM data set for input, but the data set was empty.
6nn	File open/connect	The VSAM GENCB for an RPL was not successful. For MVS or VM, the value of <i>nn</i> is determined from VSAM return codes. If register 15 contains 4, <i>nn</i> is replaced by the contents of register 0. If register 15 does not contain 4 (or 0), <i>nn</i> is replaced by 50 plus the contents of register 15.
702	File open/connect	The VSAM SHOWCB or MODCB macro was not successful. This usually means that the file is not open.
703	File open/connect	The VSAM TESTCB macro was not successful.
768	File open/connect	The OPEN or connection was not successful due to a GETMAIN error when requesting storage for control blocks associated with sequential files.
8nn	File control/request	These return codes do not have a VisualAge Generator mnemonic. Either MVS CICS returned an ILLOGIC error (if MVS CICS), or VSAM returned a return code of 8 on file input/output. The <i>nn</i> is replaced with the VSAM error code. For more information, see the <i>MVS VSAM administration Manual: Macro Instruction Reference</i> manual.
8nn	File open/connect	A storage allocation was not successful.
80C	Storage allocation	There is insufficient storage to satisfy a GETMAIN request.
802	File open/connect	The resource is not associated.
802	Storage allocation	There is insufficient storage for allocation.
803	Contents control	The module specified on a LOAD macro could not be located.
805	Contents control	The module specified on a LOAD macro is damaged.
805	Message processing	An unsupported option was specified on an INFORM macro.
806	Contents control	For MVS or VM, the module specified on a LOAD macro could not be located.
807	Contents control	Insufficient storage is available to load the specified module.

Table 78. VisualAge Generator Server Return Codes (continued)

RC	Component	Probable Cause
987	File control/request	For MVS or VM, a branch was made to the SYNAD routine as the result of a GET to a non-VSAM serial file. A possible reason is that the file is empty or the blocking factor is wrong.
988	File control/request	For MVS or VM, a branch was made to the SYNAD routine as the result of a PUT to a non-VSAM serial file. A possible reason is that the file is empty or the blocking factor is wrong. For MVS CICS or VSE CICS, a WRITE request to a spool file was not successful.
989	File control/request	An error occurred while trying to close a file.
999	File control/request	An unsupported request was made for a serial file. A probable cause is that the record associated with this file was not defined with a serial organization.

Common SQL Return Codes

Only the most frequently occurring DB2/VSE codes are listed in this section. If you receive other DB2/VSE codes or if you need a more complete explanation of one of the DB2/VSE codes, refer to the documentation for your release of DB2/VSE.

RC	Meaning
100	<p>No rows were found by SQL that meet the search criteria specified in the WHERE clause of the SQL statement, or if processing a SCAN process option in conjunction with a SETINQ or SETUPD process option, the end of the selected rows has been reached. The possible causes are the following:</p> <ul style="list-style-type: none"> • The key value(s) were not moved correctly to the host variable(s) used in the WHERE clause. • No rows meet the search criteria specified in the WHERE clause. • VisualAge Generator Developer stripped trailing blanks for the character host variables used in a LIKE predicate in the WHERE clause. You can use the /TRACE=SQLIO generation option to enable tracing of the data sent to SQL and the data coming back from SQL. See “VisualAge Generator Server Trace Facility” on page 552 for more information about using the VisualAge Generator Server trace facility.

-301, -302, -303, -304

The VisualAge Generator Developer data item definition does not match the definition of the same column in the SQL table. This can be caused by defining a column as variable length, but not defining the data item in VisualAge Generator Developer with a variable-length SQL code. This can also be caused by specifying a different length to VisualAge Generator Developer from what was defined in the SQL table.

Make the necessary changes in the VisualAge Generator Developer data item definition to match the SQL table and generate the program again.

-805

The package *owner.package.name* is not in the system.sysaccess

The package name in the message insert area returned in the SQLCA is the name of the program that issued the SQL request. The DB2/VSE package associated with the program has not been installed in the database that your program is trying to access.

-818

The DB2/VSE precompiler-generated time stamp in the load module is different from the time stamp in the package associated with the program issuing the SQL request. Both of these precompiler outputs must match and one of them is not from the most recently-run precompile. This typically happens when a program used with more than one database is modified, generated again, and precompiled, and the package produced by the precompilation has not been copied to all the databases that the program accesses. Or, the package is installed in the database and someone tries to run the previous version of the program against the database.

Ensure that you are running the most recent copy of the program.

-911, -913

A deadlock condition occurred. Possible causes are:

- The isolation level was set for repeatable read.
- There were long periods of time between commit points.
- The program issued an update process without a replace option.

Note: The program should be coded to handle these conditions.

Common VSAM Status Codes

Only the most frequently occurring VSAM codes are listed in this section. If you receive other VSAM codes or if you need a more complete explanation of one of these values, refer to the *VSE VSAM Macro Instruction Reference* manual.

OPEN request type

Code	Meaning
64	Warning message: OPEN encountered an empty alternate index that is part of an upgrade set.
74	This is a warning message indicating the data set was not properly closed. Either the implicit verify for the OPEN was unsuccessful, or the user specified that the implicit verify should not be attempted for the OPEN. A previous VSAM program might have ended abnormally. The VERIFY command of Access Method Services can be used to properly close the data set.
80	The DD statement for this access method control block is either missing or not valid.
94	Either no record for the data set to be opened was found in the available catalog or catalogs, or an unidentified error occurred while VSAM was searching the catalog.
98	Security verification was not successful; the password specified in the access method control block for a specified level of access does not match the password in the catalog for that level of access.
A0	The operands specified in the ACB or GENCB macro are inconsistent either with each other or with the information in the catalog record. You might have attempted to open an empty data set for input only (SCAN).
A8	The data set was not available for the type of processing you specified, or an attempt was made to open a reusable data set with the reset option while another user had the data set open.
BC	The data set indicated by the access method control block is not a valid type of data set for specification by an access method control block. You might have used a sequential data set as the physical file, but specified VSAM or VSAMRS as the file type for resource association when you generated the program.

C0	An unusable data set was opened for output.
C4	Access to data was requested using an empty path.

CLOSE request type

Code	Meaning
04	The data set indicated by the access method control block is already closed.
88	Not enough virtual storage was available in the address space of your program for the work area required by CLOSE.
94	An unidentified error occurred while VSAM was searching the catalog.

GET/PUT/POINT/ERASE/CHECK/ENDREQ request types

Note: The following occur when register 15=8(8).

Code	Meaning
08	An attempt is made to store a record with a duplicate key, or there is a duplicate record for an alternate index with the unique key option.
6C	The RECLLEN specified was one of the following: <ul style="list-style-type: none"> • Larger than the maximum allowed • Equal to 0 • Smaller than the sum of the length and the displacement of the key field • Not equal to the record(slot) size specified for a relative record data set
70	The KEYLEN specified was too large or equal to 0.
C0	A relative record number that is not valid was encountered.

COBOL Status Key Values

This shows the most frequently occurring COBOL status key values. If you receive other status key values or if you need a more complete explanation for one of these values, refer to the application programming language reference for your release of COBOL.

Status Key	Explanation
10	The end of a file was reached.

- 22** An attempt was made to write a record with a key that duplicated one that was already in the file.
- 23** Record not found. This can also be caused by an optional file not being allocated.
- 35** No FILEDEF command was issued for the file. This can occur if the program calls another program or transfers to another program using a DXFR statement, but the FILEDEF commands for the second program have not been added to the sample runtime REXX exec for the main program.
- 39** The physical file that you specified during resource association does not match the file characteristics that you specified during record definition. The file characteristics include file organization (sequential, relative or indexed), the prime record key, the alternate record keys, and the maximum record size.
- 44** A variable-length record was written that is not valid. This can occur if the value in the number of occurrences field is larger than the maximum value, or the value in the record length item field is larger than the maximum length of the record.
- 96** No DLBL command was issued for a VSAM file. This can occur if the program calls another program or transfers to another program using a DXFR statement, but the DLBL commands for the second program have not been added to the sample runtime REXX exec for the main program.

Chapter 59. VisualAge Generator Server Return Codes and Abend Codes for VM Systems

The information within this chapter is diagnosis, modification, or tuning information.

Return Codes

This section contains a listing of codes set by VisualAge Generator Server and returned in the COBOL return code of a program.

If a generated program completes normally, the COBOL return code is set to the value in the EZERCODE special function word. This code must be less or equal to 512. Return codes greater than 512 are reserved for VisualAge Generator Server. The return codes set by VisualAge Generator Server are:

- 693** The program ended due to an error detected by VisualAge Generator Server. The error description is reported as described in Chapter 56. Diagnosing Problems for VisualAge Generator Server on VM Systems.
- 4093** A program generated using VisualAge Generator Server Version 1.1 ended due to an error detected by VisualAge Generator Server.

If LE detects an error and returns to the operating system, the LE return code modifier (2000 - error, 3000 - severe error, or 4000 - critical error) is added to the user or VisualAge Generator Server return code.

ABEND Codes

VisualAge Generator Server reports errors by error messages whenever possible. ABENDs are issued only in situations where initialization has not progressed to the point where messages can be issued or when the error messages cannot be written to their normal destination.

- 1600** A generated program has ended because of a serious error. This occurs for one of the following reasons:
 - Storage has been corrupted so that a dump is necessary to debug the abend.
 - A severe error has occurred. Refer to the messages that were sent to ELAPRINT DDNAME.

See “VisualAge Generator Server ABEND Dumps” on page 549 for information on how to find error messages in the dump on a 1600 abend.

- 1601** A database manager or subsystem interface module was not linked with a VisualAge Generator Server program at product installation. Registers 3 and 4 in the dump contain the name of the stub program. The abending load module is the module that was not linked correctly.
- 1606** The dynamic storage stack used for working storage for VisualAge Generator Server modules was exhausted and VisualAge Generator Server could not continue.

This problem should not occur. Report the problem to the IBM Support Center.
- 1608** VisualAge Generator Server has detected a FREEMAIN request that is not valid. Collect the dump and contact the IBM Support Center for assistance.

Chapter 60. Codes from Other Products for VM Systems

The chapter contains lists of common system abend codes, COBOL and COBOL/370 runtime messages, and COBOL abend codes.

Common System Abend Codes for All Environments

Only the most frequently occurring abend codes are listed in this section. If you receive another abend code or if you need a more complete explanation of one of the abend codes, refer to the *VM/ESA System Messages and Codes* manual for your release of VM.

System 0C4

This code can occur on a DXFR statement if there is a print services or table program with the same name as the transferred-to program. This code can also occur when a print services or table program is called but there is a different program (for example, a non-VisualAge Generator program or an application program) with the same name. Using naming conventions can eliminate this problem.

This code can also occur if you add table edits to a map group that is shared by multiple programs and do not generate all the programs again.

System 0C7

Data exception. The abend occurs when fields defined as decimal or packed decimal are retrieved from a database and are found to contain data of a different format.

The abend can also occur if fields that are not initialized are used in calculations or comparisons. This happens if the program attempts to read a record from a database and the record is not found, but the program uses fields in the record anyway. To ensure that records are initialized, use a SET record EMPTY statement in the program or specify /INITADDWS and /INITRECD as generation options. Refer to the VisualAge Generator online help system for additional information

on how to initialize records using a SET record EMPTY statement in the program. Scan the helps using the phrase SET record EMPTY.

The abend can also occur when SET record EMPTY is used or when /INITADDWS and /INITRECD are used if one of the following is true:

- There are redefined records with different data types or variable field boundary alignments from the original record.
- The primary working storage record receives a transferred record that contains different data types or variable-field boundary alignments from the original record.

System 15A

Module not found in a library.

This can occur if the module is not in a library defined in your GLOBAL LOADLIB list. If the missing module name is one that you recognize as being part of your program, ensure that all of your program LOADLIBs are specified in the GLOBAL LOADLIB list.

If the missing module name is ELACxxx, the NLS language code identified by the last 3 characters of the module name is not installed on the system. This language code was specified with the /NLS generation option when the program was generated.

See “Languages Available” on page 587 for a list of valid NLS language codes.

COBOL Runtime Messages

Only the most frequently occurring COBOL runtime messages are listed in this section. If you receive other runtime messages that start with IGZ or if you need a more complete explanation of one of the messages, refer to the debugging manual for your release of COBOL.

IGZ033I **There was an attempt to call an AMODE(24) program passing a parameter address above the 16-megabyte boundary.**

Explanation: The calling program was compiled with the DATA(31) COBOL compiler option and the called program was link-edited with the AMODE(24) linkage option. The called program cannot gain access to the parameters.

IGZ075I **Inconsistencies were found in EXTERNAL file “?” in program “?”. The following file attributes did not match those of the established external file: “?”**

Explanation: This message occurs if you have used the same file name in two different programs that are being run together (for example, in a main batch program and in a program that it calls or transfers to through a DXFR statement). The message can occur in one of the following situations:

- The same record name is used in both programs, but the information you specified at resource association differs. This occurs if you

specify VSAM for a serial file when generating one program and specify SEQ for the same serial file when generating the other program.

- The same record name is used in both programs, but the definition of the record changed between the time you generated the first program and the second. For example, if you add a field and cause the record length to change or if you change the record organization, this error occurs.
- Different records that have a common file name are used in the programs. The two records have conflicting definitions, such as record length or record organization.
- Both programs display printer maps. One of the programs uses a map group that only has maps for SBCS devices. The other program uses a map group that contains some DBCS maps. You need to ensure that at least one map in the map group specifies a DBCS device type if you mix DBCS and non-DBCS maps from different map groups.

The third insert in the error message can be repeated several times to indicate all the types of differences that occurred in the definitions.

COBOL/370 Runtime Messages

Only the most frequently occurring COBOL/370 runtime messages are listed in this section. If you receive other runtime messages that start with IGZ or if you need a more complete explanation of one of the messages, refer to the *Debugging and Runtime Messages Guide* for your release of LE.

IGZ033S **An attempt was made to pass a parameter address above 16 megabytes to AMODE(24) program *program-name*.**

Explanation: An attempt was made to pass a parameter above the 16-megabyte storage line to a program in AMODE(24). The called program will not be able to address the parameter.

IGZ064S **A recursive call to active program *program-name* in compilation unit *compilation-unit* was attempted.**

Explanation: COBOL does not allow an internal program that has started to run, but has not completed, to be invoked again. For example, if internal programs A and B are siblings of a containing program, and A calls B and B calls A, this message will be issued.

IGZ066S **The length of *external data record data-record* in program *program-name* did not match the existing length of the record.**

Explanation: While processing external data records during program initialization, it was determined that an external data record was previously defined in another program in the run unit, and the length of the record as specified in the current program was not the same as the previously defined length.

IGZ075S **Inconsistencies were found in EXTERNAL file *file-name* in program *program-name*. The following file attributes did not match those of the established external file: *attribute-1 attribute-2 attribute-3 attribute-4 attribute-5 attribute-6 attribute-7***

Explanation: One or more attributes of an external file did not match between two programs that defined it.

Common COBOL Abend Codes

Only the most frequently occurring abend codes are listed in this section. If you receive another abend code or if you need a more complete explanation of one of the messages, refer to the debugging manual for your release of COBOL.

User 1015

This COBOL abend code indicates there has been a recursive call. For example, A calls B and B in turn calls A. COBOL does not support this.

User 1069

This is a COBOL abend. The VisualAge Generator program attempted to use VSAM in 31-bit mode. COBOL does not support VSAM files when the virtual machine is running in XA, ESA or XC mode. Because VisualAge Generator programs cannot be run in a 370 mode virtual machine, /FILETYPE=VSAMRS must be specified for all VSAM files when using COBOL.

User 4087

This is an LE abend code.

- If reason code is 0, the error could be due to a mismatch between the level of LE code in a saved segment and the level on the accessed LE minidisk.
- If reason code is 7, the error could be due to the region size not being large enough to run the COBOL program.

Part 14. Messages for MVS, VSE, and VM Systems

Chapter 61. VisualAge Generator Server	VisualAge Generator Server Message
Messages for MVS, VSE, and VM	Format 587
Systems 587	Reporting Problems to IBM 589
Languages Available 587	Messages 590

Chapter 61. VisualAge Generator Server Messages for MVS, VSE, and VM Systems

This chapter contains an explanation of the host message format and lists the messages that can be returned by VisualAge Generator Server when running your program or when using the parameter group utility.

The messages in the following sections are organized alphabetically by message prefix. The message numbers are listed in ascending order within each prefix. To locate the explanation and the user response for a message, first locate the message prefix, then search for the message number.

In all cases where the messages in this manual are translated into another language, use the first 8 characters of any displayed message to locate the translated version of that message within this manual.

Languages Available

The following languages are available when VisualAge Generator Server is installed on your system:

Suffix	Language
ENU	U.S. English
ENP	Uppercase English
CHS	Simplified Chinese
DEU	German
DES	Swiss German
ESP	Spanish
JPN	Japanese (Katakana)
KOR	Korean
PTB	Brazilian Portuguese

VisualAge Generator Server Message Format

Each message consists of a message identifier (for example, ELA00023P) and message text. The text is a short phrase or sentence describing the error condition.

The message identifier consists of three fields: prefix, message number, and type code. The format of the message identifier is **xxxnnnnnt**, where:

xxx Message prefix, as follows:

- ELA** These runtime messages can occur when your program stops, ends with an error, or requires special attention.
- FZE** These C++ runtime messages can occur when using the installation and print utilities FZEZREBO and FZETPRT provided with VisualAge Generator Server.
- MSG** These messages can occur when you are using the message file conversion utility.
- PRM** These messages can occur when you are using the parameter group utility.

nnnnn

Message number associated with the error condition that caused the message to be displayed

t Type code, as follows:

- I** **Information:** Indicates a minor error, such as a move from a field that is not initialized, or provides you with general information about the process you are working on. Processing continues.
- A** **Action:** Indicates that you must take some specific action before the process can continue (for example, a YES or NO response might be required). Processing continues after you complete the required action.
- P** **Problem Determination:** Indicates that a problem condition exists that requires diagnosis. Processing ends when this type of message is issued. If the problem determination message text includes a return code, see the following references for your system for an explanation of the return code:
 - **MVS Systems:** “Chapter 27. Common System Return Codes for MVS Systems” on page 251 and “VisualAge Generator Server Return Codes” on page 256
 - **VSE Systems:** “Chapter 44. Common System Return Codes for VSE Systems” on page 431 and “VisualAge Generator Server Return Codes” on page 435
 - **VM Systems:** “Chapter 58. Common System Return Codes for VM Systems” on page 565 and “VisualAge Generator Server Return Codes” on page 567

S System Action: Indicates that a system error occurred requiring you to take some action. These messages appear in English.

The message text might contain one or more inserts. When the message is displayed an insert is used to fill in names, constants, return codes, and so forth. The format of the message insert is %**xyzz**, where:

xx Number of the insert
y **C, D, or X.** These letters represent the following:
C Characters (usually a name)
D Decimal numbers (usually a length, record count, or error count)
X Hexadecimal numbers (usually a return code)
zz Length of the insert

In this manual you see messages listed like this:

```
ELA00023P Call to table program %01C07 was not successful
```

If you receive this message on your system, the insert is automatically converted. For example, if there is a problem with table program TABLNAM, the error is displayed on your system like this:

```
ELA00023P Call to table program TABLNAM was not successful
```

TABLNAM is the first insert of the message (%01) and is in character format (C) and is seven characters long (7).

Reporting Problems to IBM

Some messages indicate problems that may require the help of the IBM Support Center to solve. If you have Internet access to IBM product support, you can report the problem using these steps:

1. Go to website <http://ps.software.ibm.com>
2. Click on the link **Technical Assistance**.
3. Click on the link **Technical assistance, problem reporting and other registered services**
4. When prompted, enter your USER NAME and PIN number.
5. Follow the web instructions for reporting a new problem to IBM.

If you do not have Internet access, call the telephone number for the IBM Support Center in your country.

Messages

ELA00002P VisualAge Generator Server is required for program %01C08

Explanation: The generated COBOL program being run is not compatible with the installed version of VisualAge Generator Server.

VisualAge Generator Server ends the program with a user abend.

User Response: Contact the system administrator. VisualAge Generator Server should be installed.

ELA00003P PCB %01D03 DL/I error, function = %02C04, status code = %03C02

Explanation: The program control logic attempted a DL/I call to a teleprocessing PCB and received an error status code from IMS on the call. The message specifies the PCB that was used on the call (0 - the I/O PCB, 1 - the modifiable alternate PCB, or 2 - the express modifiable alternate PCB). It also specifies the function code and the status code. For ISRT calls, the message is accompanied by message ELA00066I, which displays the first 255 bytes of the message.

The run unit ends. If the ELASNAP data set is allocated, VisualAge Generator Server issues a SNAP dump for all status codes other than AI.

User Response: Look up the status code in the IMS messages and codes documentation for your system.

ELA00005A Date entered is not valid for defined date format %01C10

Explanation: Data entered into a map field defined with a date edit either does not meet the requirements of the format specification as defined using the map definition facility, or the month or day of the month is not valid.

It is not necessary to enter the separator characters shown in the message, but if they are

omitted, enter leading zeros. For example, if the date format is MM/DD/YY, you can enter 070491.

User Response: Enter the date in the correct format shown in the message.

ELA00007P File OPEN error on file %01C08, file status = %02C08

Explanation: The specified file did not open successfully.

The format of the file status depends on the file type.

For SEQ files, the file status is the 2-character COBOL status code followed by six zeros.

For VSAM files, the file status is composed of the 2-character COBOL status code followed by the VSAM return code (two characters), VSAM function code (one character), and the VSAM feedback code (three characters). The VSAM codes could be blank if the file OPEN was not completed.

For VSAMRS files, the file status is composed of the 2-character ACB (access control block) return code in hexadecimal format followed by six zeros.

The run unit ends.

User Response: First see the table of common COBOL and VSAM status codes in the *VisualAge Generator Server Guide for MVS, VSE, and VM*. If the codes in the message are not listed in the table, refer to the COBOL programming language reference and VSAM administration guide for your system for a definition of other file status and VSAM codes. Also look for system error messages pertaining to the specified DD name or DLBL name. Correct the error and run the program again.

ELA00008P File CLOSE error on file %01C08, file status = %02C08

Explanation: The specified file did not close successfully.

The format of the file status depends on the file type.

For SEQ files, the file status is the 2-character COBOL status code followed by six zeros.

For VSAM files, the file status is composed of the 2-character COBOL status code followed by the VSAM return code (two characters), VSAM function code (one character), and the VSAM feedback code (three characters).

For VSAMRS files, the file status is composed of the 2-character ACB (access control block) return code in hexadecimal format followed by six zeros.

The run unit ends.

User Response: First see the table of common COBOL and VSAM status codes in the *VisualAge Generator Server Guide for MVS, VSE, and VM*. If the codes in the message are not listed in the table, refer to the COBOL programming language reference and VSAM administration guide for your system for a definition of other file status and VSAM codes. Also look for system error messages pertaining to the DD name or DLBL name. Correct the error and run the program again.

ELA00009P Overflow occurred because the target item is too short

Explanation: The target of a MOVE statement or arithmetic statement is not large enough to hold the result without truncating significant digits. The value of special function word EZEORDER is 1, which specifies that the program should end if this overflow condition occurs.

The run unit ends.

In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLST or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS, VSE, and VM environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: Do one of the following:

- Increase the number of significant digits in the target data item.
- Define the program logic to handle the overflow condition using the special function words EZEORDER and EZEORDER.

ELA00014P REPLACE process option attempted without preceding UPDATE option on %01C18

Explanation: VisualAge Generator Server recognized a REPLACE process option against a record that has not been successfully read for update. This error also occurs if a REPLACE process option is associated with a specific UPDATE process option, and that process was not the one used to select the record. The read for update might have been cancelled as the result of a segmented CONVERSE process option.

The run unit ends.

User Response: Run this program with the test facility with a trace for process or group flow to determine the program logic error.

ELA00015P READ/WRITE error for file %01C08, file status = %02C08

Explanation: An I/O operation was not successful for the specified file. Program processing ends on a hard status code if the EZEORDER special function word is set to 0, or on any nonzero status code if there is not a process error routine.

The format of the file status depends on the file type.

For SEQ files, the file status is the 2-character COBOL status code followed by six zeros.

For VSAM files, the file status is composed of the 2-character COBOL status code followed by the VSAM return code (two characters), VSAM

function code (one character), and the VSAM feedback code (three characters).

The run unit ends.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLST or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS, VSE, and VM environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: First see the table of common COBOL and VSAM status codes in the *VisualAge Generator Server Guide for MVS, VSE, and VM*. If the codes in the message are not listed in the table, refer to the COBOL programming language reference and VSAM administration guide for your system for a definition of the other file status and VSAM codes. Also look for system error messages pertaining to the specified DD name or DLBL name. Correct the error and run the program again.

ELA00016P %01C08 error for file %02C08,
%03C44, file status = %04C08

Explanation: An I/O operation was not successful for the specified file. Program processing ends on a hard status code if the EZEFEFEC special function word is set to 0, or on any nonzero status code, if there is not a process error routine.

The message identifies the VSAM operation that was not successful, the VisualAge Generator file name associated with the record, the system resource name, and the file status. The file status is composed of two zeros followed by the VSAM return code (two characters), VSAM function code (one character), and the VSAM feedback code (three characters).

The run unit ends.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLST

or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS, VSE, and VM environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: First see the table of common VSAM status codes in the *VisualAge Generator Server Guide for MVS, VSE, and VM*. If the codes in the messages are not listed in the table, refer to the VSAM administration guide for your system for a definition of other VSAM codes. Also look for system error messages pertaining to the specified system resource. Correct the error and run the program again.

ELA00021I An error occurred in program
%01C08 on statement number
%02D06

Explanation: An error occurred in the specified program on the specified statement. The actual error that occurred is identified in the messages following this message.

User Response: Refer to a listing of the program, either from generation or from the print utility, to determine which statement failed. Correct the statement and generate the program again.

ELA00022P Map group format module
%01C08 could not be loaded

Explanation: The specified map group format module could not be loaded. The module is a generated object module linked as a program that contains tables that describe the format and constant fields for terminal maps in a map group. The module name is the map group name followed by the characters FM.

If the map group format module name uses the format ELAxxxFM, where xxx is the language code, the map definitions for the VisualAge Generator Server error maps could not be loaded.

The run unit ends.

User Response: Have the system administrator do one of the following:

- Verify that the specified program has been generated, compiled, and linked into a library defined in the library search order.
- For MVS/TSO environments, the search order includes the MVS/TSO called library, ISPLLIB, and STEPLIB data sets.
- For IMS/VS environments, the search order includes the STEPLIB and JOBLIB data sets.
- For MVS CICS environments, the search order includes the DFHRPL data sets.
- For VSE CICS or VSE batch, ensure that the library is in the LIBDEF search order.
- For MVS CICS or VSE CICS environments, verify that the program has been defined in the processing program table (PPT).
- On OS/2 systems, verify that the module DLL (dynamic link library) is in a directory specified in the environment variable CICSWRK.
- For the VM environment, the search order includes the logical saved segment and the GLOBAL LOADLIB list specified in ELARUN. This list includes the program LOADLIBs and the VisualAge Generator Server for MVS, VSE, and VM product LOADLIB.

ELA00023P Call to table program %01C07 was not successful

Explanation: A dynamic COBOL call to the specified table program was not successful.

The run unit ends.

User Response: Have the system administrator do one of the following:

- Verify that the specified program has been generated, compiled, and linked into a library defined in the library search order.
- For MVS/TSO environments, the search order includes the MVS/TSO called library, ISPLLIB, and STEPLIB data sets.
- For IMS/VS, IMS BMP, or MVS batch environments, the search order includes the STEPLIB and JOBLIB data sets.

- For MVS CICS environments, the search order includes the DFHRPL data sets.
- For VSE CICS or VSE batch environments, ensure that the library is in the LIBDEF search order.
- For MVS CICS and VSE CICS environments, verify that the program has been defined in the processing program table (PPT) and has the AMODE(31) program attribute.

Note: Table programs on MVS CICS and VSE CICS must be link-edited with the AMODE(31) program attribute.

- For MVS and VSE systems, if the table program named in the message is ELACxxx or ELAYYNx (where xxx and x are the NLS identifiers), verify that the customization JCL, ELACJxxx, has been run.

For MVS, VSE and VM systems, ask the person who installed VisualAge Generator Server to verify that the appropriate language (indicated by xxx or x) has been installed.

- On OS/2 systems, verify that the table DLL (dynamic link library) is in a directory specified in the environment variable CICSWRK.
- For the VM environment, the search order includes the logical saved segment and the GLOBAL LOADLIB list specified in ELARUN. This list includes the program LOADLIBs and the VisualAge Generator Server for MVS, VSE, and VM product LOADLIB.

ELA00024P Conversion table %01C08 could not be loaded

Explanation: Either the specified table program could not be loaded or the program that was loaded is not a VisualAge Generator Server conversion table.

The run unit ends.

User Response: Verify that either the correct conversion table name was specified in the generation linkage table, that a correct conversion table has been moved into the EZECONVT special function word by the program, or that a correct conversion table has

been specified on a call to EZECONV. If the table name is correct, have the system administrator verify that the conversion table program has been installed as follows:

- Verify that the specified program has been generated, compiled, and linked into a library defined in the library search order.
- For MVS/TSO environments, the search order includes the MVS/TSO called library, ISPLLIB, and STEPLIB data sets.
- For IMS/VS, IMS BMP, or MVS batch environments, the search order includes the STEPLIB and JOBLIB data sets.
- For MVS CICS environments, the search order includes the DFHRPL data sets.
- For VSE CICS or VSE batch environments, ensure that the library is in the LIBDEF search order.
- For MVS CICS and VSE CICS environments, verify that the program has been defined in the processing program table (PPT) and has the AMODE (31) program attribute.
- On OS/2 systems, verify that the table DLL (dynamic link library) is in a directory specified in the environment variable CICSWRK.
- For the VM environment, the search order includes the logical saved segment and the GLOBAL LOADLIB list specified in ELARUN. This list includes the program LOADLIBs and the VisualAge Generator Server for MVS, VSE, and VM product LOADLIB.

If the conversion table program is defined in the load library, verify that the program is using either the conversion tables shipped with VisualAge Generator Server or a table created using the conversion table format described in the section of the *VisualAge Generator Client/Server Communications Guide* on defining custom conversion tables.

ELA00026P A calculation caused a "maximum value" overflow

Explanation: During a calculation in an arithmetic statement, an intermediate result exceeded the maximum value (18 significant

digits). This condition also occurs when division by zero occurs. If the EZEORDER special function word is set to either 0 or 1, the program ends.

This error can only occur when you specify the /NUMOVFL generation option.

The run unit ends.

In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLST or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS, VSE, and VM environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: Correct the program logic either to avoid the error or to handle the error using the special function words EZEORDER and EZEORDER.

ELA00027P The data on character to numeric move is not valid

Explanation: The statement in error involves a move from a character to a numeric data item. The character data item contains nonnumeric data.

The run unit ends.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLST or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS, VSE, and VM environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: Change the program to ensure that the source operand contains valid numeric data.

ELA00029P DXFR to %01C08 was not successful

Explanation: The transfer to another program was not successful. Usually, the program being transferred to could not be found.

If the statement number at which the error occurred is zero, the error occurred after a segmented CONVERSE process option in a program that was transferred to using a DXFR statement. The program associated with the transaction code in IMS/VS or CICS is attempting to return to the program that issued the CONVERSE process option. The program that issued the CONVERSE process option has been deleted from the library since the CONVERSE process option was issued.

The run unit ends.

User Response: Have the system administrator do one of the following:

- Verify that the specified program has been generated, compiled, and linked into a library defined in the library search order.
- For MVS/TSO environments, the search order includes the MVS/TSO called library, ISPLLIB, and STEPLIB data sets.
- For IMS/VS, IMS BMP, or MVS batch environments, the search order includes the STEPLIB and JOBLIB data sets.
- For MVS CICS environments, the search order includes the DFHRPL data sets.
- For VSE CICS environments, ensure that the library is in the LIBDEF search order.
- For MVS CICS and VSE CICS environments, verify that the program has been defined in the processing program table (PPT) and has the AMODE(31) program attribute.
- On OS/2 systems, verify that the table DLL (dynamic link library) is in a directory specified in the environment variable CICSWRK.
- For the VM environment, the search order includes the logical saved segment and the GLOBAL LOADLIB list specified in ELARUN. This list includes the program LOADLIBs and

the VisualAge Generator Server for MVS, VSE, and VM product LOADLIB.

ELA00031P Call to %01C08 was not successful

Explanation: A dynamic call to the specified program was not successful.

The run unit ends.

User Response: Have the system administrator do one of the following:

- Verify that the specified program has been generated, compiled, and linked into a library defined in the library search order.
- For MVS/TSO environments, the search order includes the MVS/TSO called library, ISPLLIB, and STEPLIB data sets.
- For IMS/VS, IMS BMP, or MVS batch environments, the search order includes the STEPLIB and JOBLIB data sets.
- For MVS CICS environments, the search order includes the DFHRPL data sets.
- For VSE CICS or VSE batch environments, ensure that the library is in the LIBDEF search order.
- For MVS CICS and VSE CICS environments, verify that the program has been defined in the processing program table (PPT) and has the AMODE(31) program attribute.
- On OS/2 systems, verify that the table DLL (dynamic link library) is in a directory specified in the environment variable CICSWRK.
- For the VM environment, the search order includes the logical saved segment and the GLOBAL LOADLIB list specified in ELARUN. This list includes the program LOADLIBs and the VisualAge Generator Server for MVS, VSE, and VM product LOADLIB.

ELA00032P Called program %01C07 received a parameter list that is not valid

Explanation: A call to the specified program was not successful for one of the following reasons:

- The calling program passed too many or too few parameters.
- The parameter formats (PARMFORM) specified in the linkage table for the called and calling program differ.
- The COMMDATA parameter format was specified, and the COMMAREA passed has a different length than the length expected by the called program.

If the called program is a remote program, a CICS abend occurs. Because the COMMAREA is too small, the called program cannot notify the calling program of the error.

In all other cases, the run unit ends.

In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLST or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS, VSE, and VM environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: Verify that the argument list in the CALL statement matches the parameter list for the program being called, and then generate the called and calling program with the same linkage table specification.

ELA00033P Call to function %01C08 returned exception code %02D05.

Explanation: An exception code was returned on a call to the specified function, indicating that one of the arguments passed to the function was invalid. The REPLY option was not coded on the function call, specifying that the program should end if an exception occurs.

Refer to the function description in the VisualAge Generator reference manual for an explanation of the exception code.

The run unit ends.

User Response: The developer should fix the program so that it does not pass invalid arguments to the function.

ELA00034P Program %01C07 was defined as a main program and cannot be called

Explanation: The specified program was defined as either a main transaction program or a main batch program. It cannot be called by any other program or program.

The run unit ends.

In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLST or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS, VSE, and VM environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: If the program needs to be called, define the current program as a called program.

ELA00035A Data type error in input - enter again

Explanation: The data in the first highlighted field is not valid numeric data. The field was defined as numeric.

User Response: Enter only numeric data in this field, or press a bypass edit key to bypass the edit check. In either situation, the program continues.

ELA00036A Input minimum length error - enter again

Explanation: The data in the first highlighted field does not contain enough characters to meet the required minimum length.

User Response: Enter enough characters to meet the required minimum length, or press a bypass edit key to bypass the edit check. In either situation, the program continues.

ELA00037A Input not within defined range - enter again

Explanation: The data in the first highlighted field is not within the range of valid data defined for this item.

User Response: Enter data that conforms to the required range, or press a bypass edit key to bypass the edit check. In either situation, the program continues.

ELA00038A Table edit validity error - enter again

Explanation: The data in the first highlighted field does not meet the table edit requirement defined for the variable field.

User Response: Enter data that conforms to the table edit requirement, or press the bypass edit key to bypass the edit check. In either situation, the program continues.

ELA00039A Modulus check error on input - enter again

Explanation: The data in the first highlighted field does not meet the modulus check defined for the variable field.

User Response: Enter data that conforms to the modulus check requirements, or press a bypass edit key to bypass the edit check. In either situation, the program continues.

ELA00040A No input received for required field - enter again

Explanation: No data was typed in the field designated by the cursor. The field is required.

User Response: Enter data in this field, or press a bypass edit key to bypass the edit check. Blanks or nulls will not satisfy the data input requirement for any type of field. In addition, zeros will not satisfy the data input requirement for numeric fields. The program continues.

ELA00041P A message table prefix was not specified for a program: EZEMNO = %01C04, NLS code = %02C03

Explanation: A user message was requested either by setting the EZEMNO special function word or by specifying a user message number for a map field edit. A user message table prefix was not defined for the program.

The run unit ends.

User Response: Do one of the following:

- Add the message table prefix to the program specifications and then generate the program again.
 - Change the program to not set the EZEMNO special function word and then generate the program again.
 - Remove the user message number from the map field edit and then generate the map group again.
-

ELA00042P The expected number of inserts for message %01C08, NLS code %02C03 was not received

Explanation: The expected number of variable inserts for a VisualAge Generator Server message did not match the number received. The message text is in the language-dependent message table program, ELACxxx, where xxx is the language code.

The program is generated from the table part defined in external source format part ELACxxx. The table might have been modified and generated specifically for your installation.

The inserts show the original error message number that occurred and the language code being used. Message ELA00163P shows the original error message number that occurred and the message inserts that would have been displayed for that message.

The run unit ends.

User Response: Correct the problem identified by the original message.

If the language-dependent message table was modified, correct the modified message so that the inserts are the same as the inserts defined in the default table that was shipped with VisualAge Generator Server.

ELA00043P %01C08, %02C03

Explanation: The VisualAge Generator Server message table program ELACxxx, where xxx is the language code, did not contain a runtime message.

The program is generated from the table part defined in external source format file ELACxxx. The table might have been modified and generated specifically for your installation.

The inserts show the original error message number that occurred and the language code being used. Message ELA00163P shows the original error message number that occurred and the message inserts that would have been displayed for that message.

The run unit ends.

User Response: Correct the problem identified by the original message.

If the language-dependent message table was modified, verify that the message numbers in the modified table match the message numbers in the message table as shipped in the product sample library. Also, verify that the program loaded by the program is at the same maintenance and release level as the message table shipped in the product sample library.

ELA00044P Message %01C08, NLS code %02C03, not found

Explanation: The VisualAge Generator Server message table program ELANCxxx, where xxx is the NLS code, did not contain a runtime message. The program is generated from the table part defined in external source format file ELANTxxx. The table may have been modified and generated again at your installation.

The inserts show the original error message number that occurred and the NLS language code that was being used. The message is accompanied by message ELA00163P, which shows the original error message number that occurred and the message inserts that would have been displayed for that message.

The original error message that occurred determines if (and how) the program ends and if a SNAP dump is issued.

User Response: Correct the error identified by the first message insert.

If the message table was modified, check that the message numbers in the modified table match the message numbers in the message table as shipped in the product sample library. Also, check that the program loaded by the program is at the same maintenance and release level as the message table shipped in the product sample library.

ELA00045P Error reading message %01C08, NLS code %02C03, status %03C08

Explanation: The user message file or database did not contain a user-defined message for the language associated with the language code. Message file and databases are used only in COBOL programs generated using CSP/370 Runtime Services Version 1 Release 1.

The format of the message ID is as follows:

- Positions 1-3 = User message file
- Positions 4-8 = Message number

The status code varies depending on the type of user message file or database being used:

- For VSAM, status is eight characters. The first two bytes of code are either 08 (to specify a relative message within a record is not used) or 12 (to specify a record was not found in the VSAM file). The remaining six bytes of code are the VSAM return code (two characters), function (one character), and feedback code (three characters), all in decimal format. Refer to the VSAM administration guide for your system for a definition of the VSAM codes.
- For DL/I, status is the 2-character DL/I status code. Refer to the IMS messages and codes or the IMS application programming manuals for your system for a description of the specified status code.
- For DB2, status is the 4-character SQL code. Refer to the DB2 manuals for your system for a description of the SQL code.
- Verify that the specified program has been generated, compiled, and linked into a library defined in the library search order.
- For MVS/TSO environments, the search order includes the MVS/TSO called library, ISPLLIB, and STEPLIB data sets.
- For IMS/VS, IMS BMP, or MVS batch environments, the search order includes the STEPLIB and JOBLIB data sets.
- For MVS CICS environments, the search order includes the DFHRPL data sets.
- For VSE CICS or VSE batch environments, ensure that the library is in the LIBDEF search order.
- For MVS CICS environments, verify that the customization JCL, ELACJIC, has been run.
- For MVS CICS and VSE CICS environments, verify that the program is linked AMODE(31). Finally, verify that the print services program has been defined in the processing program table (PPT) and is enabled.
- On OS/2 systems, verify that the table DLL (dynamic link library) is in a directory specified in the environment variable CICSWRK.
- For the VM environment, the search order includes the logical saved segment and the GLOBAL LOADLIB list specified in ELARUN. This list includes the program LOADLIBs and the VisualAge Generator Server for MVS, VSE, and VM product LOADLIB.

In the IMS/VS environment, the transaction (logical unit of work) ends and processing continues with the next message. In all other environments, the run unit ends.

User Response: Have the administrator ensure that the message is defined in the program message file in one of two ways:

- Convert the message file to a VisualAge Generator message table. Generate the program and the message table again using VisualAge Generator Developer.
- If a message database is being used, add or replace the message in the message database using the Cross System Product/370 Runtime Services Version 1 Release 1 message database utility.

ELA00046P Call to print services program %01C08 was not successful

Explanation: A dynamic COBOL call to the specified print services program was not successful.

The run unit ends.

User Response: Have the system administrator do one of the following:

ELA00047P Message %01D04 was not found in message table program %02C07

Explanation: A user message could not be found in the program message table.

In all MVS, VSE, and VM environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

The run unit ends.

User Response: Either add the message to the table or modify the program to use a message that is defined in the table.

ELA00050A Number of allowable significant digits exceeded - enter again

Explanation: The user entered data into a numeric field that was defined with decimal places, a sign, currency symbol, or numeric separator edits. The number of significant digits that can be displayed within the editing criteria was exceeded by the input data; the number entered is too large. The number of significant digits cannot exceed the field length, minus the number of decimal places, minus the places required for editing characters.

User Response: Enter a number with fewer significant digits.

ELA00051P Map %01C08 was not found in map group %02C06

Explanation: The specified map name is not in the map group.

The run unit ends.

User Response: Generate the map group and the program again.

ELA00057P DELETE attempted without preceding UPDATE on record %01C18

Explanation: A DELETE process option was attempted for a record not successfully read by an UPDATE process option. The read for update might have been lost as the result of a CONVERSE process option when running in segmented mode.

The run unit ends.

User Response: Run this program with the test facility, tracing for process and statement group flow, to determine the program logic error.

ELA00061P DL/I error, function = %01C04, status code = %02C02

Explanation: DL/I returned an error status code in response to the DL/I call for the current process and either of the following occurred:

- There was no error routine specified for the process.
- Both special function words EZEFEK and EZEDLERR were set to 0 (this indicates that the program should end on abnormal DL/I conditions), and the status code specified either an abnormal condition, or a condition that was not expected.

The status code in the message comes from the DL/I PCB used for the DL/I call.

The run unit ends.

In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

This is either a program error or a database definition error.

User Response: Do the following:

1. Locate the specified error code.
 - On MVS systems, refer to the IMS messages and codes or the IMS application programming manuals for a description of the specified status code.
 - On VSE systems, refer to the DOS DL/I messages and codes manual for a description of the specified status code.
2. Correct the error.
3. Generate the program again.

ELA00062P DL/I call overlaid storage area, record %01C18

Explanation: A DL/I call read a block of data that was larger than the record defined to hold the data. The storage area immediately following the record buffer was overlaid.

The run unit ends.

In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLST

or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS and VSE environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: This is a program error. Define the record so that its length matches the length of the segment it represents and generate the program again.

ELA00063I PCB DB %01C08, segment %02C08, level %03D02, options %04C04

Explanation: This message provides additional diagnostic information for a database I/O error. The PCB passed in the DL/I call contained the specified information.

For unsuccessful DL/I I/O call, the segment name field contains the last segment along with the path to the requested segment that satisfied the call. When a program is initially scheduled, the name of the database might be put in the segment name field if no segment is satisfied.

User Response: Refer to message ELA00061P.

ELA00064I PCB key feedback area length %01D04

Explanation: This message provides additional diagnostic information for a database I/O error. The PCB passed in the DL/I call contained the specified key feedback length. This is the length of the concatenated key of the hierarchical database path.

User Response: Refer to message ELA00061P.

ELA00065I PCB key feedback area = %01C255

Explanation: This message provides additional diagnostic information for a database I/O error. The PCB passed in the DL/I call contained the specified key feedback area.

The first 255 bytes are displayed. If necessary, because of the line and data lengths, the message

wraps around to display all 255 bytes. The data is displayed as character data in the message. The message is followed by two lines that give the hexadecimal value under each character.

User Response: Refer to message ELA00061P.

ELA00066I DL/I I/O area = %01C255

Explanation: This message provides additional diagnostic information for a hard DL/I I/O error. The message displays the contents of the DL/I I/O area.

The first 255 bytes are displayed. If necessary, because of the line and data lengths, the message wraps around to display all 255 bytes. The data is displayed as character data in the message. The message is followed by two lines that give the hexadecimal value under each character.

User Response: This message is always accompanied by another message (for example, ELA00003P or ELA00061P) that specifies the error. See the explanation and user response of the accompanying message.

ELA00067I DL/I SSA %01D02: %02C255

Explanation: This message provides additional diagnostic information for a DL/I I/O error. The message displays the contents of a segment search argument (SSA) for the DL/I call. The first message insert gives the number of the SSA. The second insert gives the first 255 bytes of the SSA.

If necessary, because of the line and data lengths, the message wraps around to display all 255 bytes. The data is displayed as character data in the message. The message is followed by two lines that give the hexadecimal value under each character.

This message is repeated once for each SSA used in the DL/I call.

User Response: Refer to message ELA00061P.

ELA00068P DL/I variable segment length is not valid, segment %01C08

Explanation: A DL/I segment I/O area is shorter than the segment returned in a DL/I retrieval, or the computed segment length on an ADD or REPLACE process option is not valid.

If the process option was an INQUIRY, UPDATE, or SCAN, the BYTES parameter in the DBD is greater than the length of the record defined to VisualAge Generator.

If the process option was an ADD or REPLACE, the program has erroneously set the length of the segment. If this error occurs for a path call, the DL/I I/O area shown in message ELA00061I contains only segments before the segment with the error. Because the length is in error, the segment with the error cannot be moved to the DL/I I/O area.

The run unit ends.

In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLST or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS and VSE environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: If the error occurred in a retrieval, have the database administrator correct either the DBD or VisualAge Generator record definition, and generate the program again.

If the error occurred on an update, correct the logic associated with calculating the length of the segment. Generate the program again.

ELA00069P The value of an input variable is too large for the target SQL column

Explanation: A PACK or PACF record item in an SQL row record, that is defined as requiring an even-numbered length for SQL purposes (EVENSQ=YES), has been assigned a value that is too large to be contained within the even-numbered length.

In the IMS/VSE environment, the transaction (logical unit of work) ends and processing continues with the next message.

In all other environments, the run unit ends.

In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

In all MVS, VSE, and VM environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: Modify the program to ensure that values that overflow the even-numbered length of the record item are detected and rectified before executing any process that has an SQL row record as its object, and that uses the record item as an input host variable in its SQL statement.

This condition is not detected in programs that have the /NUMOVFL generation option; instead the high-order digit of the record item's value is truncated before being used in the SQL statement.

ELA00070P %01C04 error, status code %02C02

Explanation: DL/I returned an error status code other than QC or AL in non-VSE environments or HX or XR on VSE environments in response to a CHKP (checkpoint) or ROLB (rollback) DL/I call.

CHKP and ROLB calls are issued for the following reasons:

- The program calls the EZECOMIT and EZEROLLB service routines.
- The program ends abnormally and a PSB is active.

- The program causes a commit to be taken at a CONVERSE process option, a First Map, or because of the /SYNCXFER generation option.

The status code in the message is taken from the I/O PCB used with the DL/I call.

The run unit ends.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLSST or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS and VSE environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: Make a note of the message and notify the system programmer. On MVS systems, refer to the application programming manual or the IMS messages and codes manual for a description of the status code. On VSE systems, refer to the DOS DL/I messages and codes manual for a description of the status code.

ELA00072P %01C18, SET record SCAN not supported

Explanation: The SET SCAN indicator was on for a DL/I segment record when a SCAN process with a user-modified SSA list was used with that record. The SET SCAN indicator is not supported for DL/I calls with modified SSA lists.

The run unit ends.

User Response: Modify the program logic so that it does not set the SET SCAN indicator for a segment with a modified DL/I call.

ELA00073P SQL error, command = %01C08, SQL code = %02D04

Explanation: The SQL database manager returned an error code for an SQL statement. Program processing ends following an SQL

request whenever the SQLCODE in the SQL communications area (SQLCA) is not 0, and either of the following is true:

- No error routine is specified for the process
- The SQLCODE indicated a hard error and the EZEFEC special function word was set to 0, indicating that the program should end on SQL error conditions.

On OS/2 systems, this message is followed by the database manager message describing the SQL error code.

On all other systems, this message is followed by message ELA00074I which displays the SQLERRM data, the insert data for the DBM message describing the SQL error code.

The run unit ends.

In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLSST or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS, VSE, and VM environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: Determine the cause of the problem from the SQL code and the SQL error information.

Either correct the program or the database definition. Refer to the appropriate database manager messages and codes manual for information on the SQL code and SQL error information.

ELA00074I SQL error message: %01C70

Explanation: This message accompanies message ELA00073P when an SQL error occurs.

It displays the relational database manager error information returned in the SQLCA field SQLERRM and is repeated as many times as necessary to display the complete description.

User Response: Use the information from this message and ELA00073P to correct the error.

ELA00076P The data on character to hexadecimal move or compare is not valid

Explanation: The current statement involves either a move from a character data item to a hexadecimal data item, or a comparison between a character data item and a hexadecimal data item. The characters in the character data item all must occur in the following set for the move or compare to complete successfully:

a b c d e f A B C D E F 0 1 2 3 4 5 6 7 8 9

One or more of the characters in the character data item is not in this set. This condition causes a program error.

The run unit ends.

In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLST or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS, VSE, and VM environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: Change the program to ensure that the character data item contains valid data when the character to hexadecimal move compare operation occurs. To do so, you can use the hexadecimal map edit characteristic to ensure that input from a variable field contains valid characters.

ELA00077P Load of table %01C07 was not successful

Explanation: VisualAge Generator Server was not able to find table data associated with the table name.

The run unit ends.

User Response: Verify the table data is available.

If VisualAge Generator Developer was used to generate the table, ensure that tablename.DAT is in a directory specified in the environment variable DPATH.

If CSP/370AD Version 4 Release 1 was used to generate the table, ensure that tablename.DLL is in a directory specified in the environment variable CICSWRK.

If you are not sure which generator might have been used to generate the table, check for both of the files listed above.

ELA00078P Data file for table %01C07 is not valid

Explanation: The data file found for the indicated table is not valid.

This message is accompanied by message ELA00104I, which displays the OS/2 file name of the file in error.

The run unit ends.

User Response: Verify that the file indicated in message ELA00104I is the table data file that should be used for this table.

If this is the correct file, the file might have been corrupted. Generate the table data file using the VisualAge Generator Developer or obtain a new copy of the file from the program provider.

ELA00080A Hexadecimal data is not valid

Explanation: The data in the variable field identified by the cursor must be in hexadecimal format. One or more of the characters you entered does not occur in the following set:

a b c d e f A B C D E F 0 1 2 3 4 5 6 7 8 9

User Response: Enter only hexadecimal characters in the variable field. The characters are left-justified and padded with the character zero. Embedded blanks are not allowed.

ELA00086P %01C18 - No active SETINQ, SETUPD, or UPDATE

Explanation: A SCAN, REPLACE, or DELETE process option cannot be performed because a required process has not occurred previously. If a REPLACE process option has an associated UPDATE process option or SETUPD process option, the UPDATE or SETUPD process option must be active. One of the following conditions must exist:

- A SCAN process option for an SQL row must be preceded by a SETINQ process option or a SETUPD process option that has the same process object as the SCAN process option.
- A REPLACE process option for an SQL row must be preceded either by an UPDATE process option for the same row or by a SCAN process option for a row that was selected using the SETUPD process option.
- A DELETE process option for an SQL row must be preceded either by an UPDATE or SCAN process option for the same object.

The SCAN, REPLACE, or DELETE process option must be run in the same program that ran the previous process. All rows selected for scanning or updating are released when the program returns to the calling program.

The run unit ends.

User Response: Modify the program to issue the required process option prior to the process option that cannot be performed.

ELA00093I An error occurred in program %01C08, process or group %02C18

Explanation: An error occurred in the specified process or statement group for the specified program. Other information about the error is given in the messages that follow this message.

If a process or statement group is not active, the second insert contains the name of a section in the initialization or ending logic for the generated program.

User Response: Refer to the error messages following this message to determine the cause of the error.

ELA00096P A mixed data operand is not valid

Explanation: An operand in a move statement involving a mixed data item contains mixed double-byte and single-byte data that is not valid.

The run unit ends.

In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLSST or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS, VSE, and VM environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: Verify that all mixed strings in the move operation contain valid mixed data.

ELA00097P SID or SDB parameter is invalid

Explanation: You have specified SID or SDB parameter but the format is no correct. The correct formats are shown below:

For VSE:

SID=uuuuuuuu/pppppppp, SDB=ssssssssssssssss;
or

SID=uuuuuuuu/pppppppp SDB=ssssssssssssssss;
or

SID=uuuuuuuu/pppppppp;

For VM CMS or VM batch:

SID=uuuuuuuu/pppppppp, SDB=ssssssss
or

SID=uuuuuuuu/pppppppp

where:

uuuuuuuu

Specifies a 1- to 8- character SQL user ID

pppppppp

Specifies a 1- to 8- character SQL password

ssssssssssssssss

Specifies a 1- to 18- character database server name

ssssssss

In VM CMS or VM batch, specifies a 1- to 8-character database server name

User Response: For VSE batch, VM CMS, or VM batch systems, correct the format error for SID or SDB parameter and try the program again.

ELA00104I OS/2 file in error: %01C255

Explanation: This message provides additional information for certain error conditions. It displays the OS/2 directory path and file name for the file in error.

User Response: None required.

ELA00105I Error occurred at terminal %01C08, date %02C08, time %03C08, user %04C08

Explanation: An error occurred at the specified logical terminal on the specified date and time. This message precedes any error diagnostic information routed to an alternate error destination.

In certain situations, the values of the inserts vary as follows:

- For a batch program running in the IMS BMP or IMS/VS environments, the first variable insert is ***** when the input message queue has not yet been accessed, indicating that the terminal identifier is not known.
- For a program running in MVS batch, VSE batch, or MVS/TSO, the first insert is ***** , indicating that the terminal identifier is not known.

- For the IMS BMP, MVS batch, or VSE batch environments, the last insert (user) is the job name from the JOB statement in the JCL used to run the program.
- For the IMS/VS and CICS environments, the last insert is only provided if sign-on security is active on or provided in the system.

User Response: Examine all error messages that follow this message and precede the next occurrence of this message. Use the information from these messages to diagnose and correct the error.

ELA00106P Program %01C08 PSB does not match VisualGen PSB definition

Explanation: The PCBs passed to the program at program initialization time did not match the VisualAge Generator PSB defined for the program. The number of PCBs passed was less than the number of PCBs defined in the VisualAge Generator definition.

The run unit ends. In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLST or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS and VSE environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: Either correct the VisualAge Generator definition of the PSB and generate the program again, or correct the IMS PSB and generate it again.

ELA00109P First Map must be map %01C08, not map %02C08, for program %03C07

Explanation: The initial map expected by this program is not the map identified in the message. This error occurs when the program starts.

For the CICS, MVS/TSO, and VM environments, when another program transfers to this program, the transferring program must specify the correct map on the transfer statement.

For the IMS environment, the initial message processed for the program must be the message input descriptor (MID) for the first identified map. Instead, the second identified map was received. Either another program transferred to this program with the wrong map, or the user did not use the /FORMAT command to initiate the program.

The run unit ends.

User Response: If the error occurred when the program was started in the IMS environment, start the program again using the /FORMAT command. Otherwise, record what function you were using before the error occurred, and correct the transferring program to pass the correct map on the transfer.

ELA00110P Shared table %01C07 cannot be updated

Explanation: The program modifies a table that has been generated as a shared table. Shared tables cannot be updated.

The run unit ends.

User Response: The table must be generated as a table that is not shared or the program must be changed to not modify the table.

ELA00111P Length of First Map %01C08 is not valid

Explanation: The length of a First Map received by a program is not the length defined for the map in the program.

The run unit ends.

User Response: Generate both the program receiving the First Map and the program issuing the XFER with map statement with the same map definition for the map passed on the XFER statement.

ELA00114P A DXFR to called program %01C07 is not allowed

Explanation: Using a DXFR statement to transfer control to called programs is not allowed. For MVS CICS or VSE CICS, this message can also occur if the VisualAge Generator Server for MVS, VSE, and VM ELATSRST program has been used to initiate a called program.

The run unit ends.

In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLST or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS, VSE, and VM environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: Replace the DXFR statement with a CALL statement. If the VisualAge Generator Server ELATSRST program is being used, refer to the *Migrating Cross System Product Applications to VisualAge Generator* document for more information.

ELA00115P A DXFR to program %01C07 with First Map is not allowed

Explanation: Using a DXFR statement to transfer control to a program that has a First Map is not allowed. Refer to the *VisualAge Generator Client/Server Communications Guide* for details of the difference between the DXFR and XFER statements.

The run unit ends.

In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLST or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS, VSE, and VM environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: If the program receiving control requires a First Map, change the DXFR statement to an XFER statement. Otherwise, remove the First Map option from the transferred-to program and display the map on the first CONVERSE process option.

ELA00118P Missing PSB for program %01C07

Explanation: A PSB was specified for the named program during definition. However, the program ran as an MVS batch job, a VSE batch job, or under MVS/TSO without using a PSB. This can happen if you do not use the sample JCL or CLIST created by the generation function.

The run unit ends.

User Response: If the program contains DL/I I/O or other DL/I functions, change the runtime JCL or CLIST to run DL/I programs. If the program does not use DL/I, remove the PSB name from the program definition.

ELA00119P Programs %01C07 and %02C07 are not compatible

Explanation: A program started by a DXFR or CALL statement is not compatible with the initial program in the transaction or job for one of the following reasons:

- The program was generated for a different environment.

- The program is a main transaction with a different processing mode than the initial program (IMS/VS only).
- The program is a main transaction, and the initial program is a main batch program (IMS/VS only).
- The programs are main transactions, and the SPA options (length, ADF compatibility) specified at generation are different (IMS/VS only).

The run unit ends.

User Response: Change one or both programs to conform to the restrictions for a DXFR or CALL statement.

ELA00120P CALL to CREATX not successful, LTERM = %01C08, status code = %02C02

Explanation: The CALL was not successful resulting in the specified status code. Common status codes are:

- QH** Unknown output destination
A1 Unknown output destination.

Both status codes indicate that the 8-character logical terminal ID, shown in the first message insert, was not defined to the IMS system as either a terminal or transaction.

The run unit ends.

User Response: Do the following:

1. Ensure that the transaction code field of the record specified on the CREATX is defined to the IMS system.
2. Review the program logic ensure that the transaction code file is set correctly.
3. Refer to the IMS application programming manual or the IMS messages and codes manual for your system for an explanation of status codes other than the ones listed above.

ELA00121P CALL to AUDIT was not successful, LTERM=%01C08, status code=%02C04

Explanation: The CALL was not successful, resulting in the specified status code.

The status code is the 2-character status from the I/O PCB.

The run unit ends.

User Response: Refer to the IMS application programming manual or the IMS messages and codes manual for your system.

ELA00122P PCB for CSPTDLI call not available

Explanation: The program uses a PCB number in a CALL statement to the CSPTDLI service routine that either exceeds the number of PCBs in the PSB or references a PCB that was not passed to the program in the called parameter list.

The error can also occur in the CICS environment if the PCB number refers to PCB 0, a TP PCB, or a GSAM PCB, which are not available in CICS.

The run unit ends.

In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLST or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS and VSE environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: Either modify the call to the CSPTDLI service routine to reference a valid PCB, or modify the PSB or called parameter list definition to include the referenced PCB.

ELA00123P Basic checkpoint used in transaction-oriented BMP

Explanation: A program called the EZECOMIT service while processing as a transaction-oriented BMP. The EZECOMIT service is implemented as a basic checkpoint (CHKP) function. In the transaction-oriented BMP this resulted in a read of the message queue that overlaid program storage. The updates to the database have been committed.

This error can only occur if you are using the CSPTDLI service routine to read the message queue. The value in the EZECOMIT special function word is ignored if you use the SCAN process option to read the message queue.

The run unit ends.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLST or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS and VSE environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: Do not run the program as a transaction-oriented BMP. Alternatively, either remove the EZECOMIT special function word from the program or change the CSPTDLI service routine to a SCAN process option.

ELA00125P Error number %01D04 is not valid

Explanation: The error handler was called with an error number that it did not recognize. This is a product error.

The run unit ends.

In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLST or the dump data set of the partition. Refer to

the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS, VSE, and VM environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: Ensure that the generated COBOL code has not been modified by generating the program again. Afterwards, run the program again. If the problem persists, have the system administrator do the following:

1. Record the message number.
2. Obtain the dump.
3. Record the scenario under which this message occurs.
4. Obtain the COBOL source for the problem program.
5. Use your electronic link with IBM Service if one is available, or contact the IBM Support Center.

ELA00127P A requested function is not supported for map %01C08, map group %02C06

Explanation: A program requested a map function that is not supported for the specified map and map group. The map group was modified between the time the map group was generated and the time the program was generated. Some functions that were included in the map group (for example, help maps or the EZEMSG special function word on maps) when the program was generated, were not in the map group when the map group was generated.

The run unit ends.

User Response: Generate the map group and the program again.

ELA00128P Incompatible attributes for file = %01C08

Explanation: A program is attempting to use a GSAM file that is already opened for another program. The file characteristics (record organization, record length, fixed or variable

length records, or key specification) are defined differently for the two programs and the definitions are not compatible. The run unit ends.

If the file is EZEPRINT, the problem is due to attempting to write non-DBCS maps followed by DBCS maps.

User Response: Define the file characteristics to be the same in both programs or use a different file name for one of the programs.

ELA00129I Map %01C08 was received

Explanation: The message shows the name of a map that was received. Related messages give more information about the error.

User Response: Refer to the related error messages.

ELA00130P GSAM error, file = %01C08, function = %02C04, status code = %03C02

Explanation: An I/O error occurred on an ADD, SCAN, or CLOSE process option for a file associated with a GSAM database. Program processing ends on a hard status code if EZEFEFC is set to 0, or on any error status code if there is no process error routine.

This message can also occur on an implicit OPEN or CLSE call to the GSAM database. An implicit OPEN or CLSE call occurs as a result of an ADD or SCAN process. Program processing ends on a hard status code if EZEFEFC is set to 0, or on any error status code if there is no process error routine for the ADD or SCAN that caused the implicit OPEN or CLSE call.

An AI status code for an implicit OPEN might be caused by specifying a file name during VisualAge Generator resource association or with the ASSOCIATE command that is different from the DD name specified in the GSAM DBD.

For an ADD, message ELA00066I accompanies this message and provides the DL/I I/O area that was used for the call.

The run unit ends. If ELASNAP is allocated, VisualAge Generator Server issues a SNAP dump.

User Response: Determine the cause of the I/O error from the DL/I status code and either correct the program or the database definition. Refer to the IMS application programming manual or the IMS messages and codes manual for your system for an explanation of the DL/I status code.

ELA00131P MSGQ error, file = %01C08, function = %02C04, status code = %03C02

Explanation: An error occurred on a SCAN or ADD process for a file or a DISPLAY process for a print map when the file or EZEPRINT is associated with an IMS message queue (I/O or TP PCB). Program processing ends on a hard status code, if EZEFECEC is set to 0, or on any error status code, if there is no process error routine.

Common status codes are:

- QH** Unknown output destination (ADD, DISPLAY, or CONVERSE)
- A1** Unknown output destination (ADD, DISPLAY, or CONVERSE)
- A6** Output segment limit exceeded (ADD, DISPLAY, or CONVERSE)
- FD** Deadlock occurred (SCAN).

For an ADD, DISPLAY, or CONVERSE, the listed status codes specify that the 8-character system resource name associated with the file or EZEPRINT at generation or in the EZEDEST or EZEDESTP special function words was not defined to the IMS system as either a terminal or a transaction.

For an ADD, DISPLAY, or CONVERSE, message ELA00066I accompanies this message and shows the DL/I I/O area that was used for the call.

The run unit ends. If ELASNAP is allocated, VisualAge Generator Server issues a SNAP dump.

User Response: If the output destination is not valid, ensure that it is defined to the IMS system. Also review the program logic to ensure that EZEDEST, if used, is set correctly. For an explanation of status codes other than the ones listed above, refer to the IMS application programming manual or the IMS messages and codes manual for your system.

ELA00132P Variable length %01D02 is not valid for record %02C18

Explanation: The variable length record being written to a GSAM file or a message queue has a length that is greater than the maximum length defined for the record structure. Either the record length item contains a value greater than the maximum record length or the number of occurrences is greater than the maximum number of occurrences specified. The first message insert provides the length field that was being used. The length is the total length being written to the GSAM file, including the 2-byte length field itself, or the total length being written to the message queue, including the 12-byte header (length, ZZ field, transaction code) itself. The second message insert provides the name of a serial record being routed to a GSAM file or a message queue.

The run unit ends. If ELASNAP is allocated, VisualAge Generator Server issues a SNAP dump.

User Response: Modify the program to move a valid value into the record length item or the number of occurrences item.

ELA00134P I/O PCB conflict between programs %01C07 and %02C07

Explanation: A program invoked using CALL or DXFR accesses the I/O PCB as a serial file. The initial program in the transaction is a main transaction and the current program accesses the I/O PCB. The control logic for a main program cannot operate correctly when a program that is transferred to via a DXFR or is called accesses the I/O PCB.

The run unit ends.

User Response: Modify the program so it does not access the I/O PCB or invoke the program from a main batch program.

ELA00135P The program is not expecting a First Map

Explanation: The program is a main transaction program with no First Map defined. Another program attempted to transfer with a map to this program.

The run unit ends.

User Response: Either change the transferring program to transfer only a record to the specified program, or change the specified program to specify First Map.

ELA00136P DL/I error occurred in work database operation

Explanation: An error occurred during use of the work database when it was implemented using DL/I. This message is accompanied by additional DL/I diagnostic messages, including ELA00061P, that provide additional information about the error. Message ELA00061P includes the DL/I function and status code. Refer to the IMS messages and codes or IMS application programming manual for your system for a description of the status code.

The run unit ends. If ELASNAP is allocated, VisualAge Generator Server issues a SNAP dump.

User Response: This is a database definition error or an error in the definition of the work database PCB in your IMS PSB. Record this information and any other diagnostic messages, and notify the system administrator.

ELA00137P SQL error occurred in work database operation

Explanation: An error occurred during use of the work database when it was implemented using SQL. This message is accompanied by additional SQL diagnostic messages, including ELA00073P, that provide additional information about the error.

The run unit ends. If ELASNAP is allocated, VisualAge Generator Server issues a SNAP dump.

User Response: Determine the cause of the problem from the SQL code and the SQL error information in related message ELA00074I, and correct the database definition.

ELA00138P %01C07 was replaced in the middle of a conversation

Explanation: The specified program was generated as segmented. During a CONVERSE process option, the program was replaced in the load library or directory between the time the map was written to the terminal and the time the user entered data. The program conversation with the user started with the original version of the program and cannot be resumed.

The run unit ends. In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

User Response: Run the program again.

ELA00139P MFS map program %01C06 and MFS map %02C08 have different versions

Explanation: An MFS mapping services program attempted to process a message input descriptor for an MFS map that was generated at a different time than the MFS mapping services program. Both the MFS mapping services program and the map it works with must be built in the same generation step.

This is probably a problem with the installation of either the program or the MFS map after generation of a map group. One of the following might have occurred:

- The MFS mapping services program might have been compiled and linked without installing the MFS maps, or vice versa.
- The MFS map might have been installed in an MFS test library, but you did not enter an IMS /TEST MFS command prior to starting the transaction.

- The MFS map might have been installed in the MFS production library, and you entered a /TEST MFS command prior to starting the transaction.
- The MFS map might have been used in an XFER with a map from another program. The transfer-from program used a different map group, but the map name on the XFER is the same as the First Map name for the transfer-to program.

In the IMS/VS environment, the transaction (logical unit of work) ends and processing continues with the next message. In all other environments, the run unit ends.

User Response: Ensure that the same version of the MFS mapping services program and the MFS control blocks are installed in the correct libraries. If an XFER and First Map are involved, ensure that the transfer-from and transfer-to programs use the same map group.

ELA00140P Segmentation storage size discrepancy for %01C07

Explanation: The size of the segmentation storage record is not valid for the specified program.

Possible causes for the error include:

- The program is replaced in the load library or directory in the middle of a program conversation with the user.
- The program performs an XFER statement with a map to a First Map program when the programs were generated with different map definitions.
- The program performs a segmented CONVERSE process option and the EZESEGTR special function word contains a transaction code that is associated with a program that has no relationship to the program performing the CONVERSE process option. If the EZESEGTR special function word is used to switch transaction codes, the new transaction must start either the same program that was started by the old transaction or the program that issued the CONVERSE process option.

The run unit ends.

In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLST or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS and VSE environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: Try the transaction again. If the program works correctly, the error was caused by a relink in the middle of the conversation. If the error still occurs, determine why there is a mismatch and correct the situation that caused the error.

ELA00141P Table %01C07 cannot be modified. Delete %02D06 bytes.

Explanation: The generated table has a length greater than 65535 bytes. The current program modifies the table. Shared tables longer than 65535 bytes cannot be modified in CICS environments, due to shared storage size constraints in CICS.

The run unit ends.

User Response: Either change the logic of the program so that the table is not modified, or decrease the size of the table contents by the specified number of bytes.

ELA00142P Map %01C08 in group %02C06 not supported on this device

Explanation: A map has been sent to a device using IMS Message Format Services, but the device type was not in the list of devices specified for the map using the VisualAge Generator device selection function. The message appears when either of the following occurs:

- A printer map was sent to a destination that is defined as a terminal in the IMS System Generation. The destination is the system resource name specified for EZEPRINT at generation or an override value loaded into the EZEDESTP special function word at run time. The message appears at the terminal where the printer map was directed, not at the terminal that originated the transaction. Program processing continues.
- A terminal map is defined in a map group that contains multiple maps with different device selections. The device to which the map was directed was not specified using the VisualAge Generator device selection function. The message appears at the terminal that originated the transaction as the result of a CONVERSE or an XFER with a map. The program conversation with the user at this device ends because there is no way for the user to enter data. The program continues processing with the next input message on the message queue.

The program is not notified by MFS that a problem has occurred. Therefore, message ELA00142P is built into the MFS source to provide a method of notifying you when an error occurs. A SNAP dump is not issued.

User Response: If the error occurred for a printer map, review the resource association information specified during generation, the program logic used to set the value of EZEDESTP, and the MFS generation options (/MFSDEV, /MFSIGNORE, and /MFSEATTR) to determine the appropriate corrections to make. Depending on the corrections required, generate either the program or map group again. In addition, if the printer map was sent to a terminal device, it might be necessary for the system administrator to purge the messages pending for the terminal using the IMS /DEQ command.

If the error occurred for a terminal map, review the terminal device types specified for this map and the MFS generation options (/MFSDEV, /MFSIGNORE, and /MFSEATTR) to determine the appropriate corrections to make. Generate the map group again.

If the program using the terminal map is a nonconversational program (/SPA=0 generation option), the user only needs to clear the screen and type another transaction code to resume work.

If the program that used the terminal map is a conversational program (/SPA generation option greater than 0), the user must clear the screen, type /EXIT to end the conversation and then type another transaction code to resume work.

ELA00143P Table %01C07 is not a message table

Explanation: A message table was specified for the program. The table specified is not a message table.

The run unit ends.

User Response: Either define the table as a message table and generate the table again, or correct the message table name specified for the program and generate the program again.

ELA00144P Segmentation storage error

Explanation: Segmentation storage has an internal error mapping memory.

The run unit ends. In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

In all MVS and VSE environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: This is an internal system error. Contact the system administrator for assistance.

ELA00145A Map name required - enter /FOR %01C06O map-name

Explanation: The map group has more than one map, but a valid map name was not entered when the IMS /FOR command was used to display the map.

User Response: Enter the /FOR command again, using the following format:

ELA00146P Segmentation status error

Explanation: The status byte for segmentation storage management is lost and the program has no way to recover.

This error occurs when a PA key is pressed prior to pressing the ENTER key or a PF key for an IMS conversational transaction.

If the program was generated with /SPA=n or /SPA=(n,ADF), then there was no recovery feature generated in the program.

If the program was generated with /SPA=(n,m) or /SPA=(n,ADF,m) then the recovery feature was generated in the program, but was bypassed. A bypass of the recovery feature occurs when a deferred message switch comes from a non-VisualAge Generator program or a VisualAge Generator-generated program that was not generated with the same /SPA generation option.

In the IMS/VS environment, the transaction (logical unit of work) ends and processing continues with the next message.

User Response: Restart the transaction sequence and avoid using PA keys while on a VisualAge Generator-generated screen.

Consider generating the VisualAge Generator programs with one of the /SPA generation options that will allow recovery from pressing a PA key.

ELA00147A Key sequence is not valid. Last screen will display - enter the data again

Explanation: A PA key was pressed prior to pressing the ENTER key or a PF key. IMS has reserved the use of the PA keys. All modifications on the previous screen are lost.

User Response: Enter the data again and avoid use of PA keys while on a VisualAge Generator-generated screen.

ELA00149I %01C07 command ignored during message database load

Explanation: The PSB for the message database specifies that the database is being initially loaded. Only ADD commands are supported during initial load of a DL/I message database.

User Response: Run the message utility again, specifying the PSB for the database.

ELA00151P %01C07 of message record to / from message database failed

Explanation: The message utility program encountered an error inserting or deleting a message in the message database. This message is accompanied by either DL/I or SQL diagnostic messages describing the error.

If an ELASNAP DD statement is specified in the JCL, Runtime Services issues a snap dump. The run unit ends.

User Response: Review the diagnostic messages. Verify that the database has been successfully defined by checking either the DL/I DBD assembly or the DB2 message database create job (ELAMSJL2) messages. Correct the problem and run the job again.

ELA00152I Message file %01C03 has been added

Explanation: The indicated user message file has been successfully added to the message database.

User Response: Test the programs that use this user message file.

ELA00153P %01C08 failed on file %02C08

Explanation: While running the message utility, an attempt was made to access (open, close, read, or write) the indicated file. The access failed and the message utility ended. The first message insert indicates the type of access that failed. The most common errors are a missing DD card for the file or DCB parameters that are not correct..

User Response: Refer to the job listing for system error messages pertaining to the indicated DD name. Correct the error and run the job again, starting with the command that caused the error.

ELA00154I Message file %01C03 has been replaced

Explanation: The indicated user message file has been successfully replaced in the message database.

User Response: Test the programs that use this user message file.

ELA00155I Message file %01C03 has been deleted

Explanation: The indicated user message file has been successfully deleted from the message database.

User Response: Change the program using this user message file to use another message file and generate the program again.

ELA00156I Replace on non-existent message file %01C03, file was added

Explanation: A REPLACE command was issued for the indicated message file, but the file did not exist in the message database. The file was added instead.

User Response: None, provided the file was added to the correct message database.

ELA00157P %01C08 failed on file %02C08, file status = %03C06

Explanation: While running of the message utility, an attempt was made to access (open, close, read, or write) the indicated VSAM file. The file identifies the DD name. The file status consists of the VSAM return code (2 characters), function (1 character), and feedback code (3 characters). The access failed and the message utility terminated. The first message insert indicates that type of access that failed.

User Response: Refer to the VSAM administration guide for your system for a definition of the status codes. Also look at the job listing for system error messages pertaining to the indicated DD name. Correct the error and run the job again, starting with the command that caused the error.

ELA00158P Syntax error on command

Explanation: A command being processed by the message utility did not follow the correct syntax. The message utility ends.

User Response: Correct the command and rerun the job, starting with the command that had the incorrect syntax.

ELA00159P Message file %01C03 already exists in the message database

Explanation: An attempt to add a user message file failed because the message file already existed in the message database for the language specified in the current message utility command. The return code is set to 08.

User Response: Use the REPLACE command to update the message file in the message database.

ELA00160P Message file %01C03 does not exist in the message database

Explanation: An attempt to remove or list a user message file failed because the message file does not exist in the message database for the language specified in the current message utility command. The return code is set to 08. If the insert is an asterisk, you attempted to list all messages in an empty message database.

User Response: Correct the message file ID in the command and run the job again.

ELA00162P Message I/O error, type %01C04, file %02C08, code %03C08

Explanation: An error occurred when a program generated using Cross System Product/370 Runtime Services Version 1 Release 1 attempted to open or close a user message file.

The type variable insert specifies VSAM as the message file type. The file insert specifies the DD name. The first two bytes of the code insert are either 08 (to specify an OPEN) or 16 (to specify a CLOSE). The next two bytes are the ACB (Access control block) return code in hexadecimal format. The remaining bytes in the code insert are zero.

The run unit ends.

User Response: Have the administrator do one of the following:

- Determine the cause of the problem from the VSAM error code. First see the table of common VSAM codes in the *VisualAge Generator Server Guide for MVS, VSE, and VM*. If the codes are not listed in the table, refer to the VSAM administration guide for your system for a definition of other VSAM codes. Also verify that the user message file is allocated correctly.
- Convert the message file to a message table and generate the program again under VisualAge Generator Developer or CSP/370AD Version 4 Release 1.

ELA00163P %01C08, %02C60

Explanation: This message is used when a VisualAge Generator Server message cannot be found in the language-dependent message table program ELACxxx, where xxx is the language code.

The first variable insert in this message is the error message number for the error that actually occurred. The second insert in this message contains one of the message inserts that is used by the error that actually occurred. This message is repeated as many times as necessary to report all inserts. The inserts are reported in order by their number: %01, %02, and so on.

User Response: See the message with the corresponding message number in this manual. Take the action appropriate for that message. Also, contact the system administrator to determine why the message could not be found in the VisualAge Generator Server language-dependent message table program.

ELA00164P %01C08, %02C04, %03C02, %04X08

Explanation: The error handler was not successful in using a DL/I call to write diagnostic information about another error to normal destinations for error information. The variable inserts contain the following information:

- Destination from the terminal identifier field of the PCB used in the call.
The destination can be the error destination specified at program generation, the user terminal ID, or the IMS log.
- DL/I function
- DL/I status code
- PCB Address

VisualAge Generator Server ends the program with a user abend.

User Response: See the section on diagnosing problems in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for information about locating the diagnostic messages in the dump. These messages relate to the original error that ended the program. Also verify that the ERRDEST value specified at generation is included in the IMS system generation.

ELA00166P The recursion stack exceeds the maximum size allowed

Explanation: The stack that contains information to support recursion or segmentation has become too large.

The run unit ends.

In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

In all MVS, VSE, and VM environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: Check for an infinite loop that is causing a large number of recursions. Either limit the number of recursions, or reduce the number of processes in the program.

ELA00167I The diagnostic message queue is empty

Explanation: The diagnostic print utility for IMS ended without printing any diagnostic messages because the queue was empty.

User Response: None required.

ELA00168P %01C03

Explanation: The NLS language code in the file allocated to ELAMSG as shown in the insert is not valid. The VisualAge Generator Server utility ends because the language code for messages and report headings cannot be determined.

User Response: Correct the JCL so that the ELAMSG DD statement references a sequential file or in-stream data that contains a valid NLS code in columns 1 through 3 of the first record. Refer to “Languages Available” on page 587 for a list of the valid NLS codes.

ELA00169I Work database purged of %01D08 records older than day %02C06, time %03C06

Explanation: The utility that purges obsolete records from the work database has completed normally.

User Response: None required.

ELA00170P Input is not valid

Explanation: Either the date or the time provided to the utility that purges obsolete records from the work database was nonnumeric or was not valid.

The run unit ends.

User Response: Ensure that the date is in Julian format (YYDDD - two positions for the year and three positions for the day of the year). Ensure that the time is in HHMMSS format (two position for the hour, two positions for the minutes, and two positions for the seconds). The date and time specified must be at least 24 hours before the time that the purge program is run.

ELA00171A The value is truncated to fit on the host system

Explanation: The value typed in the field marked by the cursor is too long to be converted to the format required for storing mixed strings on the host system. The value is truncated to the length that can be successfully stored in the host system format.

An ASCII value that can be converted without truncation must have at least two blanks at the end of the field for each DBCS string within a mixed string.

User Response: None required. You can modify the truncated value and then press Enter.

ELA00172I CICS error, system identifier %01C08

Explanation: An error occurred on a CICS function to be performed on a remote system. The message displays the CICS identifier for the remote system.

This message is always issued along with other messages that identify the function being performed and the CICS error return information.

User Response: None required.

ELA00173P An error occurred in remote program %01C08, date %02C08, time %03C08

Explanation: An error occurred in a remote program that caused the remote program to stop running. Diagnostic messages might have been logged at the remote location giving information about the error. The date and time stamp on this message can be used to associate the messages logged at the remote system with this error message.

The run unit ends.

User Response: Report the error to the system administrator.

ELA00174P %01C08 cannot be used in called programs on a remote system

Explanation: The EZECOMIT and EZEROLLB special function words and the COMMIT and RESET service routines cannot be used in a remote called batch program or in a program called by a remote called batch program.

The run unit ends.

User Response: Move the EZECOMIT, EZEROLLB, COMMIT, or RESET call to the program that called the remote program. Calls to the COMMIT service routine and RESET service routine must be converted to calls using the EZECOMIT special function word and the EZEROLLB special function word before being moved to the calling program.

ELA00179P An error occurred starting transaction %01C08

Explanation: IMS or CICS indicates that an error occurred when a program attempted to start the specified transaction. A message following this message gives the IMS or CICS error codes.

The run unit ends.

User Response: Determine the cause of the error from the following message and correct the error.

ELA00180P Error recovery PCBs not passed to program

Explanation: The program was called by a non-VisualAge Generator program. Two required PCBs (PCB 0 and 2) were not passed to the program. The PCBs are required for issuing rollback and commit functions, and reporting error conditions.

The error results in an abend with a dump because the PCBs for reporting and recovering from the error are not available.

The run unit ends.

User Response: Modify the program to pass PCB 0 and 2 to the program using either the EZEDLPSB or EZEDLPCB special function words as parameters.

ELA00181P PCB 0 not passed to program %01C07

Explanation: The program was called by a non-VisualAge Generator program. PCB 0 was not passed to the program. This PCB is required for issuing rollback and commit functions, and reporting error conditions.

The run unit ends.

User Response: Modify the program to pass PCB 0 to the program using either the EZEDLPSB or EZEDLPCB special function words as parameters.

ELA00183P SYNCPOINT not allowed with PCB parameters

Explanation: The program called EZECOMIT, EZEROLLB, COMMIT, or RESET service routines. Each of these functions results in a CICS SYNCPOINT command, which ends PSB scheduling. Either this program or a program that called this program had the EZEDLPCB special function word in the called parameter list. The PCB address passed in the parameter list is no longer valid because the PSB is not active.

The run unit ends.

User Response: Either modify the program so it does not issue a commit or rollback, or modify the program to receive the EZEDLPSB special function word as a parameter instead of the EZEDLPCB special function word.

ELA00184P Program %01C07 and mapping services program %02C08 are not compatible

Explanation: The specified program and mapping services program are generated for different systems.

The run unit ends.

User Response: Generate the mapping services program for the same environment as the program.

ELA00185P Length of %01D02 for record %02C18 is not valid and conversion ended

Explanation: Conversion of a variable length record between the workstation format and host format cannot be performed because of one of the following conditions:

- The record length for the current record indicates that the record ends in one of the following:
 - The middle of a numeric (BIN, NUM, NUMC, PACK, or PACF) field
 - The middle of a DBCS character
 - The middle of an SO/SI string.
- The record is longer than the maximum length defined for the record.

The run unit ends. In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

In all MVS, VSE, and VM environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: Modify the program to set the record length so that it ends on a valid field boundary.

ELA00186P A mixed string in a conversion operation is not valid

Explanation: Conversion of a mixed field from EBCDIC to ASCII or from ASCII to EBCDIC cannot be performed because the double-byte data value is not valid.

The run unit ends. In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

In all MVS, VSE, and VM environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: Modify the program to ensure that the records to be converted contain valid mixed data.

ELA00187P Conversion table %01C08 does not support DBCS character conversion

Explanation: Conversion of a mixed or DBCS field from ASCII to EBCDIC or EBCDIC to ASCII cannot be performed because the specified conversion table does not include conversion tables for double-byte characters.

The run unit ends.

In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLST or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS, VSE, and VM environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: Modify the program to specify a conversion table that contains the double-byte conversion tables valid for the DBCS or mixed data being converted. Refer to the *VisualAge Generator Client/Server Communications Guide* for more information.

ELA00188P Conversion Error. Function: %01C25, Return Code: %02C05, Table: %03C08

Explanation: A system function was called to perform code page conversion for data used in a client/server program. The function failed.

Possible causes for the failure are:

- The code pages identified in the conversion table are not supported by the conversion functions on your system.
- For DBCS conversion where the source data is in ASCII format, the source data was created under a different DBCS code page than the code page that is currently in effect on the system.

User Response: Correct the cause of the error.

ELA00191I Program %01C07, generation date %02C08, time %03C08

Explanation: An error in the specified program has occurred. The error is identified in other messages preceding this message. The error might be caused by changes to individually generated components of the program.

User Response: Verify the generation date and time of the program with that of other generated components.

ELA00192I Print services program %01C08, generation date %02C08, time %03C08

Explanation: An error in the specified print services program has occurred. The error is identified in other messages preceding this message. The error might be caused by changes to individually generated components of the controlling program.

User Response: Verify the generation date and time of the print services program with that of other generated components in the program.

ELA00193I MFS for map %01C08, generation date %02C08, time %03C08

Explanation: An error using the Message Format Services (MFS) definitions generated for the specified map has occurred. The error is identified in other messages following this message. The error might be caused by changes to individually generated components of the controlling program.

User Response: Verify the generation date and time of the MFS definitions with that of other generated components in the program.

ELA00195I Map group format module %01C08, generation date %02C08, time %03C08

Explanation: An error in the specified map group format module has occurred. The error is identified in other messages preceding this message. The error might be caused by changes to individually generated components of the controlling program.

User Response: Verify the generation date and time of the map group format module with that of other generated components in the program.

ELA00201P MVS %01C08 error in service %02C08, RC = %03D04

Explanation: VisualAge Generator Server received an error return from an MVS macro. The inserts identify the macro name, the VisualAge Generator Server program name, and the return code.

The run unit ends.

In all MVS and VM environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: Contact the system administrator. Refer to the TSO/E documentation for more information on the MVS/TSO commands and return codes. Use the return codes to determine the cause of the error.

ELA00202P The file name %01C65 in EZEDEST or EZEDESTP is not valid

Explanation: The name in one of the special function words EZEDEST or EZEDESTP is not in a valid format. This message can be triggered by a spool file name which has a format that is not valid or, for VSAMRS files on VSE batch only, when a value other than a DLBL name is moved to the EZEDEST special function word.

The run unit ends.

User Response: Refer to the VisualAge Generator Developer online helps for the valid syntax. Correct the program and generate it again.

ELA00203P CICS I/O error on file %01C08, resource %02C08

Explanation: The current program has attempted to gain access to a CICS file, and CICS returned a status code that indicated an I/O error occurred. The file is the logical file name specified in the record definition. The resource is the CICS file control table (FCT) or destination control table (DCT) name.

Possible causes of the error are the following:

- The file does not exist on disk.
- The file is not defined in the CICS FCT or DCT.
- The file was specified to be opened when first referenced.
- On MVS CICS, the file was closed using the CSMT or CEMT transactions.
- For MVS CICS, the DD statement for the file in the CICS startup JCL either is missing, does not match the FCT name, or is in error.
- For VSE CICS, the DLBL statement for the file in the CICS startup JCL either is missing, does not match the FCT name, or is in error.
- The file has been changed or otherwise corrupted.

Message ELA00204I is also displayed with the information from the EXEC interface block (EIB).

The run unit ends. VisualAge Generator Server issues a dump based on information supplied for the transaction with the diagnostic controller utility.

User Response: Have the CICS administrator refer to the CICS messages and codes manual for an explanation of the EIB codes. Correct the error and then run the program again.

ELA00204I CICS EIBFN %01X04, RCODE %02X12, RESP %03D04, RESP2 %04D04

Explanation: The current program has received an error code for a CICS command. The run unit ends.

User Response: Refer to the CICS application programmers' guide for an explanation of the EXEC interface block (EIB) codes. Correct the error and then run the program again.

ELA00205P A CICS %01C22 error occurred in service %02C08

Explanation: VisualAge Generator Server received an error status code for a CICS command. This message identifies the command and the service program that issued the command. This message is accompanied by message ELA00204I, which contains the response codes from the EXEC interface block (EIB).

The run unit ends. VisualAge Generator Server issues a dump based on information supplied for the transaction with the diagnostic controller utility.

User Response: Have the system administrator use the CICS diagnostic information in this message and in message ELA00204I to determine the cause of the error. Correct the error and then run the program again.

ELA00206P Format of file %01C08 is not valid, reason code %02C01, resource %03C56

Explanation: The attributes of the system resource associated with the specified file name are not compatible with the attributes defined for the record in the program. The reason code identifies one of the following attributes:

- Key offset
- Key length
- Access method
- Record format (SEQRS file type only)
- Record length

An access method mismatch occurs when the type of data set allocated does not match what the program expects. For example, a VSAM file is allocated as a system sequential file or a partitioned data set is allocated as a sequential file without specifying a part name.

The run unit ends.

User Response: Change the record definition, the resource association, or both, so that the record attributes match the system resource attributes. Generate and test the affected programs again.

ELA00207P The attributes for file %01C08 are not compatible, reason code %02C01

Explanation: A program has attempted to use a file having file attributes that differ from another program within the run unit. All programs within a run unit must use the same attributes for a file. The reason code identifies one of the following attributes:

- Key offset
- Key length
- Access method
- Record format
- Record length
- Use of the EZELOC special function word to identify the location of a remote file

The run unit ends.

User Response: Change the record definitions, the resource associations, or both, so that all programs within a run unit have identical attributes for the file. Generate and test the affected programs again.

ELA00208P Print services program %01C06 and map group format module %02C08 were generated separately

Explanation: The specified print services program attempted to process a map that was generated at a time different from the map group format module. Both the print services program

and the map group format module must be generated at the same time.

The run unit ends.

User Response: Ensure that the print services program and the map group format module were generated at the same time and are installed in the correct libraries.

ELA00209I Backout completed successfully after abnormal termination for transaction %01C04

Explanation: The specified CICS transaction ended abnormally with the code specified in accompanying message ELA00222P. VisualAge Generator Server termination was successful in backing out all changes to recoverable resources and closing all open external resources associated with the transaction.

User Response: No action required.

ELA00210P Service number %01D04 is not valid

Explanation: An attempt was made to start a VisualAge Generator Server routine that does not exist or that is not valid.

The run unit ends.

In CICS environments, VisualAge Generator Server issues a dump based on options selected using the diagnostic controller utility.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLSST or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS, VSE, and VM environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: Ensure the generated COBOL code has not been modified by generating the program again. Afterwards, try the program again. If the problem persists, have the system administrator do the following:

1. Record the message number.
2. Obtain the dump.
3. Record the scenario under which this message occurs.
4. Obtain the COBOL source for the problem program.
5. Use your electronic link with IBM Service if one is available, or contact the IBM Support Center.

ELA00212P Error encountered gaining access to file %01C08, spool resource %02C65

Explanation: An error was received when attempting to gain access to a spool file. The message is accompanied by message ELA00204I, that contains response codes from the CICS EXEC interface block (EIB).

If the function was a write spool request (EIBFN 5602) and the spool resource name was specified as node ID without being qualified by user ID, then an error will occur if the user did not log on using the CICS logon procedure.

The run unit ends. VisualAge Generator Server issues a dump based on information supplied for the transaction with the diagnostic controller utility.

User Response: If the spool resource name specifies node ID without specifying user ID, log on using the CICS logon procedure before running the program again. Otherwise, refer to the CICS customization documentation for an explanation of the codes that are returned by the spool interface and correct the problem specified in the response codes.

Refer to the documentation provided with the VisualAge Generator Developer product for additional information on the system resource name format.

ELA00213I VSE/POWER return code %01X01, VSE/POWER feedback code %02X01, XPCCB return code %03X01

Explanation: The current program has received an error code for a VSE/POWER command.

The run unit ends.

User Response: Refer to the VSE/POWER application programming guide for an explanation of the VSE/POWER codes. The VSE/POWER return code is the hexadecimal value of PXPXRETC in the VSE/POWER reply buffer. The VSE/POWER feedback code is the hexadecimal value of PXPFBKCD in the VSE/POWER reply buffer. The XPCCB return code is the value of IJBXRETC in the VSE/POWER XPCCB.

ELA00215P PSB does not match VisualAge Generator PSB definition

Explanation: The PCBs passed to the program at program initialization time did not match the VisualAge Generator PSB defined for the program. The number of PCBs passed was less than the number of PCBs defined in the VisualAge Generator definition. This message is accompanied by ELA00217I.

The run unit ends. VisualAge Generator Server issues a dump based on information supplied for the transaction with the diagnostic controller utility.

User Response: Either correct the DL/I PSB or correct the VisualAge Generator PSB definition, and generate the program again.

ELA00216P CICS DL/I error, function %01C04, UIBFCTR %02X02, UIBDLTR %03X02

Explanation: CICS detected an error in a DL/I call. The message variable inserts specify the function being requested and the return codes from the CICS user interface block (UIB). If the function code is PCB, the program was attempting to schedule the program PSB. The message is accompanied by message ELA00217I.

Common return codes are as follows:

UIBFCTR	UIBDLTR	Description
08	00	Argument on DL/I call not valid. This error can occur if the IMSESA installation option in module ELARPIOP is specified as YES, but the IMS environment is not IMS/ESA.
08	01	PSB not found. The PSB must be defined to CICS.
08	03	The calling program has already successfully issued a scheduling (PCB) call that has not been followed by a TERM call.
08	05	PSB initialization was not successful.
08	06	The PSB in the scheduling call is not defined in the program control table (DLZACT).
08	07	A TERM call was issued when the task had already been terminated.
08	09	An MPS batch program attempted to issue a PCB call for a read-only PSB or for a nonexclusive PSB if program isolation was active.
08	FF	DL/I not active
0C	02	Intent scheduling conflict

The run unit ends.

User Response: Check the definition of the call to the CSPTDLI service routine in the program, if

the DL/I call is not valid. Otherwise, correct the problem specified by the error code. For additional codes, refer to the CICS application programmers' guide for your system to determine the meaning of the error codes.

ELA00217I Program %01C07, PSB name %02C08

Explanation: An error was detected in the specified DL/I program. The message is accompanied by messages ELA00215P or ELA00216P, which identify the problem.

The run unit ends.

User Response: Refer to the accompanying messages for the problem cause.

ELA00218P CALL to AUDIT not successful, journal id = %01D05, journal type = %02C02

Explanation: In CICS, a CALL statement to the AUDIT service routine was not successful. This message is accompanied by ELA00204I, which displays the contents of EIBRESP.

Common EIBRESP codes for CICS are as follows:

22 LENGERR

The computed length for the journal record exceeds the total buffer space allocated for the journal data set as specified in the journal control table (JCT) entry for the data set

43 JIDERR

Occurs if the specified journal identifier does not exist in the JCT

The run unit ends.

User Response: Refer to the CICS programming documentation for your system in order to define journal data sets, or contact the system administrator.

ELA00219P %01C22 error for %02C06 file
%03C08, %04C56

Explanation: An I/O operation was not successful for the specified file. Program processing ends on a hard status code if the EZEFECE special function word is set to 0, or on any nonzero status code if there is no process error routine.

The message identifies the I/O operation, the file type (SEQRS, VSAMRS, or SPOOL), the VisualAge Generator logical file name, and the system resource name associated with the file. On VSE batch, the message is accompanied by message ELA00213I that contains the return and feedback codes from VSE/POWER for SPOOL file types.

The run unit ends.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLST or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

In all MVS, VSE, and VM environments, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

User Response: Check that the correct data set has been allocated for this file.

ELA00220P Dynamic allocation was not successful, file %01C08, return %02D04, error reason code %03X04.

Explanation: VisualAge Generator Server was not successful on an attempt to perform dynamic allocation for the specified file. The other inserts are the return code in register 15 and the error reason code returned by the SVC 99 instruction. The most common cause is that the file was not available. If you want your program to receive control after getting a File Not Available (FNA) status, you must specify a process error routine

and set EZEFECE to 1. If either condition is not met, VisualAge Generator Server will end your program.

The run unit ends.

User Response: Contact the system administrator. Refer to the MVS System Programming: System Macros and Facilities manual for an explanation of the codes.

ELA00221P File %01C08, system resource name %02C56, not found

Explanation: VisualAge Generator Server attempted to dynamically allocate the file with the system resource name shown in the message. The file could not be found. If the system resource name is a 1- to 8-character DD name, then there is no DD card for the file in the job JCL, or there was no ALLOCATE command for the DD name in the MVS/TSO CLIST. If the system resource name is a data set name, then the data set either does not exist or is not cataloged.

The run unit ends.

User Response: If the name is a DD name, allocate a file to the DD name in the JCL or CLIST. If the name is a data set name, ensure that the file exists and is cataloged.

ELA00222P Transaction %01C04 ended abnormally with CICS abend code %02C04

Explanation: The specified CICS transaction ended abnormally with the specified code.

On MVS CICS systems, the following additional information is provided:

- On CICS Version 2 systems, if the ABEND code is ASRA or ASRB, this message is accompanied by the message ELA00223P and the ABEND exit can determine the module within which the error occurred.
- On later CICS systems, if the abend code is ASRA or ASRB, CICS message DFHAP0001 identifies the offset in the module at which the error occurred. The diagnostic control option specified for transaction abends using the

VisualAge Generator Server diagnostic control utility determines whether a dump occurs.

On VSE CICS systems, if the ABEND code is ASRA or ASRB, this message is accompanied by ELA00223P and the ABEND exit can determine the module within which the error occurred.

On CICS OS/2 systems, if the program is successfully able to rollback SQL changes and close COBOL files, this message is accompanied by message ELA00209I.

The VisualAge Generator Server abend handler ends the program by issuing another ABEND command using the same code.

User Response: See "ABEND Codes" in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for a description of abend codes using the format ELAx. Refer to CICS or user program documentation for an explanation of other abend codes.

On CICS OS/2 systems, if message ELA00209I does not accompany this message, bring down CICS OS/2 to ensure that OS/2 cleans up any database and file connections left open.

ELA00223P Program %01C08 abended at offset %02X08

Explanation: The specified program has abended with an ASRA or ASRB abend code. This indicates that a program check has occurred at the specified hexadecimal offset.

VisualAge Generator Server ends the program with a user abend.

User Response: If the program is a generated COBOL program, use the compile listing to find the COBOL verb that was running when the program ended abnormally. The COBOL comments identify the VisualAge Generator statements associated with the COBOL verb. Determine from the dump whether the problem was caused by bad data passed to the program. If the generated COBOL is in error, use your electronic link with IBM Service if one is available, or contact the IBM Support Center.

If the program is a non-VisualAge Generator program, use the compile listing, the dump, or both to find the problem. If the program is a VisualAge Generator Server module, use your electronic link with IBM Service if one is available, or contact the IBM Support Center.

ELA00225P Temporary storage queue name %01C08 is not valid

Explanation: The special function word EZEDEST is set to a temporary storage queue name that is not valid. The name conflicts with a queue name that is reserved for VisualAge Generator. Names cannot begin with EZE.

The run unit ends.

User Response: Specify a valid temporary storage queue name in the program.

ELA00228P The program attempted to use the resource %01C65 with file %02C07 and file %03C07

Explanation: The program attempted to associate the same system resource with two different files. The resource cannot be associated with two different files at the same time.

The run unit ends.

User Response: Examine the program and correct the logic. Generate and test the affected programs again.

ELA00229P CALL to CREATX was not successful, TRANSID = %01C04, TERMID = %02C08

Explanation: In CICS, a CALL statement to the CREATX service routine was not successful. This message is accompanied by the message ELA00204I, which displays the contents of EIBRESP.

Common codes are as follows:

- | | |
|-----------|---|
| 11 | TERMID error |
| | The terminal specified on the CREATX call is not known to CICS. |
| 28 | TRANSID error |

The transaction specified on the CREATX call is not known to CICS.

The run unit ends.

User Response: Have the system administrator define the terminal or transaction to CICS.

ELA00230P An error was encountered accessing CICS queue %01C08

Explanation: An error was received when attempting to access a CICS queue. The queue can be a transient data queue or temporary storage queue. This message is accompanied by message ELA00204I, which contains response codes from the CICS EXEC interface block (EIB).

The run unit ends. VisualAge Generator Server issues a dump based on information supplied for the transaction with the diagnostic controller utility.

User Response: Refer to the CICS application programmers' guide for an explanation of the response codes.

ELA00231P Error encountered retrieving data passed to program %01C08

Explanation: An error was received when attempting to retrieve data being passed to this program by either an XFER statement with a record or a CALL statement to the CREATX service routine. This message is accompanied by message ELA00204I, which contains response codes from the CICS EXEC interface block (EIB).

The run unit ends. VisualAge Generator Server issues a dump based on information supplied for the transaction with the diagnostic controller utility.

User Response: Refer to the CICS application programmers' guide for an explanation of the codes that are returned.

ELA00232P Map %01C08 in map group %02C06 is not defined or is not supported

Explanation: The specified map does not exist or is not defined for the type of device being used.

The run unit ends.

User Response: Either define the map for your device type or select the device for the map. Generate the map group again.

If you are running on a CICS system, have the system administrator check that the alternate screen size for your device type is specified in the PCT entry for your transaction.

If the map group name uses the format ELAxxx, where xxx is the language code, the map group might have been modified incorrectly. The ELAxxx map group contains the VisualAge Generator Server error maps.

ELA00233P %01C08 error on file %02C08, status bytes: %03C01 %04C03

Explanation: An I/O operation failed for the specified file. It is possible that the file is being used by another process. This message specifies the COBOL statement performed and the file associated with the definition. The COBOL/2 status key 1 and status key 2 fields are also displayed. If the value of status key 1 is 9, the status key 2 field is converted from binary to decimal before it is displayed.

This message is accompanied by message ELA00104I, which displays the OS/2 file name of the file in error. The run unit ends.

User Response: Use the appropriate COBOL/2 publication to diagnose the error, and take the recommended corrective action.

ELA00237P CICS TS Queue %01X16 error occurred in work database operation for program %02C07

Explanation: An error was received when attempting to access a CICS temporary storage queue. This message is accompanied by message

ELA00204I, which contains response codes from the CICS EXEC interface block (EIB).

If the error is an INVREQ (EIBRESP=16), the problem might be caused by VisualAge Generator Server attempting to write a record that is longer than the control interval size for the VSAM data sets used for the auxiliary storage queue. The maximum segmentation record size written by VisualAge Generator Server is set by the TSQUE option in the installation options module ELARPIOP. TSQUE specifies the maximum size as the number of kilobytes; the default value is 16 KB.

The run unit ends.

User Response: Refer to the CICS application programmers' guide for an explanation of the codes. If the control interval size is the problem, have the system administrator assemble the installation module again after setting the TSQUE value to a value less than the control interval size. Refer to the Server for MVS, VSE, and VM program directory for your system for more information.

ELA00239P Map program %01C08 cannot support print request from program %02C08

Explanation: A program and print services program were generated with a different /MSP generation option. The print services program does not contain the type of printer support (GSAM or SEQ) required by the program.

The run unit ends.

User Response: Generate the map group again with the MSP print support required by the program.

ELA00241I A new SQL plan was created for program %01C08

Explanation: The plan for this program was not found by the database manager or the plan that existed was not valid. A file with the name of the program and BND as the extension was located, and a valid plan was created from the BND file.

The program continues.

User Response: None required.

ELA00242P Unable to create plan because file %01C08.BND is not valid

Explanation: An attempt was made to create a valid plan for a program. The BND file for the program either was not found or was not valid.

If the BND file was not valid, the actual OS/2 DBM message describing the SQL error is displayed following this message. The absence of additional messages indicates that the BND file was not found while searching the CICSWRK path. The run unit ends.

User Response: Ensure that the BND file for the program is valid and available in one of the directories specified in the environment variable CICSWRK.

ELA00243P The database manager could not be started

Explanation: The database manager was not active when the program attempted to access an SQL resource. The attempt to start the database manager was unsuccessful. The run unit ends.

User Response: Ensure that the database manager is installed on your workstation and that the SQLLIB directory is specified in the LIBPATH statement in your CONFIG.SYS file.

ELA00245P SQL database %01C18 could not be started

Explanation: The attempt to start the database was unsuccessful.

The actual OS/2 DBM message describing the SQL error is displayed following this message. The run unit ends.

User Response: Refer to the OS/2 documentation to determine what action should be taken to recover the database.

ELA00249P Mapping services program %01C08 compiled with DATA(31) cannot be used by program

Explanation: A mapping services program compiled with the DATA(31) compiler option has been loaded for a program link-edited as AMODE(24).

User Response: Compile the mapping services program again with the COBOL DATA(24) option. Also ensure the /DATA=24 generation option is specified whenever the map group is generated.

ELA00250P Program cannot process data with 31-bit addresses

Explanation: The initial program in the run unit was compiled with DATA(31). The current program was link-edited as AMODE(24). This is not compatible.

User Response: Do one of the following:

- Compile the initial program in the run unit as DATA(24).
- Link-edit the current program as AMODE(31).

ELA00251P Table %01C08 compiled with DATA(31) cannot be used by program

Explanation: A table compiled with the DATA(31) compiler option has been loaded for a program link-edited as AMODE(24).

User Response: Compile the table program again with the COBOL DATA(24) option. Also ensure the /DATA=24 generation option is specified whenever the table is generated.

ELA00252P Error on file %01C08, queue name %02C08, RC = %03C08

Explanation: An I/O logic error was detected by VisualAge Generator Server during processing of an I/O process option for a CICS temporary storage queue. Either the program did not have an error routine for the process or the EZEFEFC special function word was set to end the program on hard I/O errors.

Because the error was detected by VisualAge Generator Server instead of the access method, the return code value consists of the characters RS (for runtime services) followed by a VisualAge Generator return code number.

The run unit ends. VisualAge Generator Server issues a dump based on information supplied for the transaction with the diagnostic controller utility.

User Response: See the section on return codes in the *VisualAge Generator Server Guide for MVS, VSE, and VM* to determine the meaning of the VisualAge Generator return code, and take the appropriate action.

ELA00253P Program %01C08 was not generated to receive map %02C08

Explanation: The specified program received a map as input, but the program does not contain processing logic for handling segmented programs. Either the wrong transaction name was specified when the program was started, or the wrong program was specified in the definition of the transaction.

The program was started as a result of one of the following:

- Specifying the transaction code for the program on a segmented CONVERSE process option instead of the transaction code for the program that ran the CONVERSE process option.
- In IMS/VS, using the /FORMAT command for a map that specifies the transaction code for the program.

The program must include either a First Map specification or segmented CONVERSE process option for the map being received.

The run unit ends.

User Response: Either specify the correct transaction code on the CONVERSE process option or an XFER with map statement, or specify the correct First Map in the receiving program. Generate the modified program again.

ELA00254P Invalid AUDIT parameters, journal ID = %01D05, type = %02C02, length = %03D05

Explanation: One or more parameters supplied for the AUDIT service routine are not within acceptable ranges. The journal ID must be between 1 and 99. The type must be in the range X'A0' to X'FF'. The minimum length is 28. The maximum length is 28 plus the size of the largest record used in a CALL statement to the AUDIT service routine within the program.

The run unit ends. VisualAge Generator Server issues a dump based on information supplied for the transaction with the diagnostic controller utility.

User Response: Correct the error and generate the program again.

ELA00255P AUDIT parameters are not valid, type = %01C02, length = %02D05

Explanation: One or more parameters supplied for the AUDIT service routine are not within acceptable ranges. The type must be in the range X'A0' to X'FF'. The length must be between 28 and 32765.

The run unit ends.

In the VSE batch environment, VisualAge Generator Server issues a SNAP dump that is directed either to the system logical unit SYSLST or the dump data set of the partition. Refer to the section on controlling error reporting in the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information.

User Response: Correct the error and generate the program again.

ELA00300I A new copy was requested for part %01C08

Explanation: A new copy was requested for the programs associated with the specified part. Newly started transactions use the new copy of the program.

User Response: None required.

ELA00301I The diagnostic control options were changed

Explanation: The diagnostic control options were changed after a user request from the VisualAge Generator Server Diagnostic Control utility.

User Response: None required.

ELA00302I Error message queue sent to print destination

Explanation: The contents of the transient data queue containing the error messages were sent to the spooling system after a user request from the VisualAge Generator Server Diagnostic Print utility.

User Response: None required.

ELA00303I Error message queue sent to print destination and deleted

Explanation: The contents of the transient data queue containing the error messages were sent to the spooling system after a user request from the VisualAge Generator Server Diagnostic Print utility. The contents of the transient data queue were then deleted.

User Response: None required.

ELA00304A Type a valid selection number, then Press Enter.

Explanation: The selection number entered for a field on one of the VisualAge Generator Server utility panels is not valid. The cursor is positioned at the field in error.

User Response: Type a valid selection, then press Enter.

ELA00305A Type a name. Then Enter.

Explanation: A required field was left blank on one of the VisualAge Generator Server utility panels. The cursor is positioned at the empty field.

User Response: Type a valid name, then press Enter.

ELA00306P CICS new copy was not successful for program %01C08. Press F2.

Explanation: The CICS SET NEWCOPY command was not successful for the specified part. The specified part was requested on the VisualAge Generator Server New Copy panel.

User Response: Press F2 to view message ELA00204I, which contains the CICS response information from the EXEC interface block (EIB). Verify that the part name is correct. Refer to the CICS application programmers' guide for an explanation of the EXEC interface block (EIB) codes.

ELA00308P I/O error on error message queue. Press F2.

Explanation: A CICS error occurred when attempting to gain access to the error destination queue identified on the VisualAge Generator Server Diagnostic Print panel.

User Response: Press F2 to view message ELA00204I, which contains the CICS response information from the EXEC interface block (EIB). Verify that the error destination name is correct. Refer to the CICS application programmers' guide for an explanation of the EXEC interface block (EIB) codes.

ELA00309A Error message queue was not found

Explanation: The error destination queue identified on the VisualAge Generator Server Diagnostic Print panel was not found.

User Response: Specify the correct error destination queue name on the panel.

ELA00310A Type a valid response, then press Enter.

Explanation: A value that was not recognized was specified in the field where the cursor is positioned. Valid values are shown following the field on the map.

User Response: Type a valid value in the field, then press Enter.

ELA00313I Default options are in effect for this transaction

Explanation: You made a request to view the diagnostic control options in effect for a specific transaction. The options currently in effect for the transaction are the default options.

User Response: To exit, press F3. To change the options for this transaction do the following:

1. Type the new options.
2. Select action 1.
3. Press Enter.

ELA00314I Error message queue was empty

Explanation: A request was made to print an error message queue that does not contain any messages.

User Response: None required.

ELA00315I Trace transaction list was updated successfully

Explanation: The list of transactions you specified to be traced has been processed successfully.

User Response: None required.

ELA00316I Trace filter criteria updated successfully

Explanation: The list of trace filter criteria you specified has been processed successfully.

User Response: None required.

ELA00317P Service number is not valid

Explanation: The trace filter criteria contains a service number that is not valid. If this error is detected during ELATRACE data set parsing on MVS/TSO, IMS BMP, MVS batch, VSE batch, VM CMS, or VM batch, the run unit ends.

User Response: Do one of the following:

- For MVS batch, VSE batch, IMS BMP, MVS/TSO, VM CMS, or VM batch, correct the service number specification in the ELATRACE data set and run the program again.
- For IMS/VS or CICS programs, correct the service number.

ELA00318P Tag in %01C08 is not valid

Explanation: The filter criteria contains a tag that is not valid. Valid tags are FILTER, EFILTER, APPLS, EAPPLS, SERVICES, and ESERVICES. The run unit ends.

User Response: Correct the tag specification and run the program again.

ELA00319P Missing or misplaced tag in %01C08

Explanation: The filter criteria contains a missing or misplaced tag. The run unit ends.

User Response: Correct the filter criteria and run the program again.

ELA00320P Too many programs in %01C08

Explanation: The filter criteria contains too many programs. The maximum number is 16. The run unit ends.

User Response: Reduce the number of programs or remove all program filter criteria, then run the program again.

ELA00321P Too many services in %01C08

Explanation: The filter criteria contains too many services. The maximum number is 32. The run unit ends.

User Response: Reduce the number of services or remove all service filter criteria, then run the program again.

ELA00322P One or more filters has a invalid value.

Explanation: One or more codes entered for the DATASTREAM, TRACETOFILE, APPSTMT, SQLIO, SQLERR or IDUMP filters is not valid. The valid code that is entered must be either Y (yes) or N (no).

For MVS/TSO, IMS BMP, MVS batch, VSE batch, VM CMS, and VM batch, the run unit ends.

If you are defining filters online on IMS, MVS CICS, or VSE CICS, the filter containing the value that is not correct is highlighted.

User Response: Do one of the following:

- For MVS/TSO, IMS BMP, MVS batch, VSE batch, VM CMS, and VM batch environments, specify either Y or N for these filters and run the program again.
- For IMS/VS or CICS environments, type one of the valid values for the highlighted filter as shown on the map, then press Enter.

ELA00323P I/O error on storage queue %01C08. Press F2.

Explanation: An error was received when attempting to access a temporary storage queue in the diagnostic message print utility. Press F2 to view message ELA00204I, which contains response codes from the CICS EXEC interface block (EIB).

User Response: Refer to the CICS application programmers' guide for an explanation of the codes.

ELA00324P Error reading trace control record. Press F2.

Explanation: An error was encountered when attempting to read or write to the trace control record in CICS. Press F2 to view more information.

For MVS CICS or VSE CICS, message ELA00204I is displayed, which contains response codes from the CICS EXEC interface block (EIB).

For CICS OS/2, a message is displayed showing the OS/2 system call that was not successful and its return code.

User Response: Have the system administrator determine the cause of the problem from the accompanying error messages.

ELA00325P Error opening %01C08

Explanation: An error was encountered when attempting to open the specified data set or the input device.

User Response: Do one of the following:

- For MVS batch, IMS BMP, MVS/TSO, VM CMS, or VM batch, ensure that the data set has the correct attributes for the specified data set.
- For VSE batch, ensure that the system logical unit SYSIPT is correctly assigned in the JCL.

ELA00326P Error reading %01C08

Explanation: An error was encountered when attempting to read the specified data set or input device.

User Response: Do one of the following:

- For MVS batch, IMS BMP, or MVS/TSO, ensure that the data set has the correct attributes for the specified data set.
- For VSE batch, ensure that the system logical unit SYSIPT is correctly assigned in the JCL.

ELA00332P Error %01D03 opening trace file

Explanation: The CSP/2RS trace facility was unable to open trace file ELATRACE.DAT.

User Response: Use the error return code in the message to determine the cause.

ELA00333P Error %01D03 repositioning trace file

Explanation: The CSP/2RS trace facility was unable to advance the pointer for trace file ELATRACE.DAT.

User Response: Use the error return code in the message to determine the cause.

ELA00336P Error %01D03 writing to trace file

Explanation: The CSP/2RS trace facility was unable to write to trace file ELATRACE.DAT.

User Response: Use the error return code in the message to determine the cause.

ELA00337P Error %01D03 getting shared segment for trace facility

Explanation: The CSP/2RS trace facility received a bad return code on a DosGetShrSeg.

User Response: Use the error return code in the message to determine the cause.

ELA00338P Error %01D03 allocating shared segment for trace facility

Explanation: The CSP/2RS trace facility received a bad return code on a DosAllocShrSeg.

User Response: Use the error return code in the message to determine the cause.

ELA00342A The maximum number of copies already exists for the table

Explanation: The maximum number of copies of a table that can be used in a CICS region at one time is 5. The request for a new copy of the table was rejected.

User Response: Old copies of a table that are in use are freed when all the transactions using the table end. Try the new copy request again later.

ELA00343P Error %01D03 creating trace semaphore

Explanation: The CSP/2RS trace facility received an error return code when attempting to use a semaphore to serialize access to the trace file. The function was DosCreateSem.

User Response: Use the error return code in the message to determine the cause.

ELA00344P Error %01D03 opening trace semaphore

Explanation: The CSP/2RS trace facility received an error return code when attempting to use a semaphore to serialize access to the trace file. The function was DosOpenSem.

User Response: Use the error return code in the message to determine the cause.

ELA00345P Error %01D03 requesting trace semaphore

Explanation: The CSP/2RS trace facility received an error return code when attempting to use a semaphore to serialize access to the trace file. The function was DosSemRequest.

User Response: Use the error return code in the message to determine the cause.

ELA00346P Error %01D03 clearing trace semaphore

Explanation: The CSP/2RS trace facility received an error return code when attempting to use a semaphore to serialize access to the trace file. The function was DosSemClear.

User Response: Use the error return code in the message to determine the cause.

ELA00347P Error %01D03 closing trace semaphore

Explanation: The CSP/2RS trace facility received an error return code when attempting to use a semaphore to serialize access to the trace file. The function was DosCloseSem.

User Response: Use the error return code in the message to determine the cause.

ELA00361P Dynamic allocation failed, file %01C08, return code %02D04, from the %03C07 command

Explanation: While attempting to do a dynamic allocation for the specified file, VisualAge Generator Server for VM received the displayed return code from a FILEDEF or DLBL command. This is most likely caused by the specification of a file mode that is not valid.

User Response: Refer to the error messages preceding this message for more information about the cause of the problem.

ELA00362P VM CMS %01C17 error in service %02C08, RC = %03D04

Explanation: VisualAge Generator Server received an error return from a VM CMS macro. The inserts identify the macro, the VisualAge Generator Server program name, and the return code.

In the VM environment, VisualAge Generator Server issues a SNAP dump if the ELASNAP data set is allocated.

If the ELASNAP data set is allocated, VisualAge Generator Server issues a SNAP dump.

The run unit ends.

User Response: Contact the system administrator. The CMS macros are documented in the *VM/ESA CMS Application Development Reference for Assembler* manual. Use the return codes to determine the cause of the error.

ELA00363P An incompatible terminal configuration change has been detected

Explanation: VisualAge Generator Server detected a change to a terminal that is different from the previous terminal on which the program was running. Changing terminal configurations while a program is running is not supported.

The run unit ends.

User Response: Restart the program.

ELA00364I Snap dump is in progress

Explanation: This is an informational message which is displayed on the screen to inform you that a problem has occurred and that a snap dump is being taken.

User Response: Please be patient. The snap dump could take a while. When the snap dump is complete, a VisualAge Generator Server error panel is generally displayed with messages indicating what went wrong.

ELA00402E Preparation for *application_name* failed - Return code: *prep_retcode*

Explanation: The preparation REXX exec called the ELAPREP EXEC to process the specified part. The ELAPREP EXEC generated a non-zero return code.

User Response: Refer to the preceding error messages to determine the cause of this error.

ELA00410E An incorrect parameter was passed to *exec_filename exec_filetype*. The parameter was *parm*

Explanation: An input parameter to the specified routine was incorrect.

User Response: Correct the indicated parameter and rerun the preparation REXX exec.

ELA00411E A value for parameter *MBR=* has not been specified. It is a required parameter

Explanation: The ELAPREP EXEC and the ELASQLAA EXEC require a non-null value to be specified for the *MBR=* parameter. The *MBR=* parameter indicates the member to be processed. The generation process should have built the preparation REXX exec with the correct value for this parameter.

User Response: Generate the application again to ensure the *MBR=* parameter in the preparation

REXX exec is not modified. Then try rerunning the preparation REXX exec. If the problem persists, contact IBM Software Support.

ELA00412E Value *parameter_value* for parameter *parameter_name* failed validation. It must be length 1 to 8

Explanation: The ELAPREP EXEC detected an error in the value of the specified parameter.

User Response: Refer to the description of the parameter in the HELP screen for ELAPREP (enter HELP ELA) or see “Chapter 54. Execs for Preparing and Running Programs on VM Systems” on page 517 for more information. Correct the indicated parameter and rerun the preparation REXX exec.

ELA00413E {Null value | The value *VMFMODE_parm_value*} was specified for *VMFMODE=* {*parameter* | and is not valid}. It must be a single alphabetic character

Explanation: The value specified for *VMFMODE=* was not valid.

User Response: Refer to the description of the parameter in the HELP screen for ELAPREP or ELASQLAA (enter HELP ELA) or see “Chapter 54. Execs for Preparing and Running Programs on VM Systems” on page 517 for more information. Correct the indicated parameter and rerun the preparation REXX exec.

ELA00415E Value *parm_value* in *parm_value_list* for parameter *parm_name* failed validation. It must be length 1 to 8

Explanation: The ELAPREP EXEC detected an error in the value of the specified parameter.

User Response: Refer to the description of the parameter in the HELP screen for ELAPREP (enter HELP ELA) or see “Chapter 54. Execs for Preparing and Running Programs on VM

Systems” on page 517 for more information. Correct the indicated parameter and rerun the preparation REXX exec.

ELA00416E {Null value was specified for DATA= parameter | Value DATA_parm_value for parameter DATA= failed validation}. It must be 24 or 31

Explanation: The value of the DATA= parameter of the ELAPREP EXEC must be 24 or 31.

User Response: Refer to the description of the parameter in the HELP screen for ELAPREP (enter HELP ELA) or see “Chapter 54. Execs for Preparing and Running Programs on VM Systems” on page 517 for more information. Correct the indicated parameter and rerun the preparation REXX exec.

ELA00418E {Null value was specified for COBLIST= parameter | Value COBLIST_parm_value for parameter COBLIST= failed validation}. It must be either DISK, PRINT or NOPRINT

Explanation: The value for the COBLIST= parameter of the ELAPREP EXEC must be DISK, PRINT, or NOPRINT.

User Response: Refer to the description of the parameter in the HELP screen for ELAPREP (enter HELP ELA) or see the “Chapter 54. Execs for Preparing and Running Programs on VM Systems” on page 517 for more information. Correct the indicated parameter and rerun the preparation REXX exec.

ELA00419E {Null value was specified for parm_name parameter | Value parm_value for parameter parm_name failed validation}. It must be either Y or N

Explanation: The specified parameter requires a Y for Yes or an N for No.

User Response: Refer to the description of the parameter in the HELP screen for ELAPREP or ELASQLAA (enter HELP ELA) or see “Chapter 54. Execs for Preparing and Running Programs on VM Systems” on page 517 for more information. Correct the indicated parameter and rerun the preparation REXX exec.

ELA00420E {Null value was specified for ENV= parameter | Value ENV_parm_value for parameter ENV= failed validation}. It must be either VMCMS or VMBATCH

Explanation: The value of the ENV= parameter of the ELAPREP EXEC must be either VMCMS or VMBATCH. The generation process should have built the preparation REXX exec with the correct value for this parameter.

User Response: Generate the application again to ensure that the ENV= parameter in the preparation REXX exec is not modified. Then rerun the preparation REXX exec. If the problem persists, contact IBM Software Support.

ELA00421E {Null value was specified for PARTTYPE= parameter | Value PARTTYPE_parm_value for parameter PARTTYPE= failed validation}. This exec only processes either EZESRC or EZEFMT

Explanation: The value of the PARTTYPE= parameter for the ELAPREP EXEC must be either EZESRC or EZEFMT. The generation process should have built the preparation REXX exec with the correct value for this parameter.

User Response: Generate the application again to ensure that the PARTTYPE= parameter in the preparation REXX exec is not modified. Then rerun the preparation REXX exec. If the problem persists, contact IBM Software Support.

ELA00422E File mode *filemode* on VMFMODE= is not accessed [in R/W mode]

Explanation: ELAPREP and ELASQLAA require that the file mode specified for the VMFMODE= parameter be accessed in read/write mode prior to their invocation.

User Response: Use the CMS QUERY ACCESSED command to review the current file modes that are accessed. Enter the VM commands to access the correct resource in read/write mode at the file mode specified by VMFMODE=.

ELA00423E Value *parm_value* for parameter *parm_name*= failed validation. It must be either a VMLINK nickname, a CMS minidisk specified as USERID/VDEV, or an SFS directory name

Explanation: The indicated parameter requires a valid location be specified. A location is a VMLINK nickname, a CMS minidisk, or an SFS directory.

User Response: Refer to the description of the parameter in the HELP screen for ELAPREP or ELASQLAA (enter HELP ELA) or see "Chapter 54. Execs for Preparing and Running Programs on VM Systems" on page 517 for more information. Also refer to the syntax definition of a location in the HELP screen. Correct the indicated parameter and rerun the preparation REXX exec.

ELA00424E Error from ELASQLAA EXEC - RC = *rc*

Explanation: The ELAPREP EXEC determined that the part currently being processed contained SQL and called ELASQLAA to preform SQL preprocessing. ELASQLAA generated a non-zero return code.

User Response: Refer to the preceding error messages to determine the cause of this error.

ELA00425E RC = *rc* from GLOBAL MACLIB command

Explanation: A CMS GLOBAL MACLIB command was issued and generated a non-zero return code. If ELAPREP issued this message, it specified the VisualAge Generator Server MACLIB (ELAVMPSI) and the standard COBOL MACLIB. The standard COBOL MACLIB varies depending on the COBOL product that you are using and was specified during VisualAge Generator Server installation.

User Response: Refer to the online HELP information for the CMS GLOBAL command for information on the return code value. If you require more help to correct this problem, ask the person who installed VisualAge Generator Server to ensure that the ELAVMPSI MACLIB exists at the location specified for 'Location of the VisualAge Generator Server MACLIB' field when the ELAVMSET EXEC was invoked. Also ensure that the standard COBOL MACLIB exists at the location specified for 'Location of the COBOL Compiler' field. Correct the problem using ELAVMSET and rerun the preparation REXX exec.

ELA00426E CMS STATE error RC = *rc* for input file *filename filetype filemode*

Explanation: A CMS STATE command was issued for the specified file and generated a non-zero return code. If ELAPREP issued this message, it was validating the existence of the input COBOL source file or the input LINKCTL file or the exec specified by the USEREXECL= parameter.

User Response: Refer to the online HELP information for the CMS STATE command. The file mode for the COBOL source file or the input LINKCTL file is the value that was specified for the ELAPREP VMFMODE= parameter. The USEREXECL= exec can be on any accessed file mode. Ensure that the specified file exists. If necessary, generate the application again and ensure all the files associated with the application are successfully uploaded to the VM system. Then rerun the preparation REXX exec.

ELA00427E COBOL compile return code is rc

Explanation: The COBOL compiler was called and generated a non-zero return code.

User Response: Refer to the COBOL listing file for compiler error messages. The ELAPREP COBLIST= parameter controls the generation of the COBOL listing file. The user-defined symbolic parameter COBLIST can be used at generation time to set COBLIST= in the preparation REXX exec.

ELA00428E RC = rc from GLOBAL TXTLIB command

Explanation: A CMS GLOBAL TXTLIB command was issued and generated a non-zero return code. If ELAPREP issued this message, the command was issued for the standard COBOL TXTLIB and the user TXTLIB that was specified for the ELAPREP USERLKED= parameter. The standard COBOL TXTLIB varies depending on the COBOL product that you are using and what was specified during VisualAge Generator Server installation.

User Response: Refer to the online HELP information for the CMS GLOBAL command for information on the return code value. Ensure that the TXTLIBs defined with USERLKED= are on an accessed file mode. If you require more help to correct this problem, ask the person who installed VisualAge Generator Server to ensure that the standard COBOL TXTLIB exists on the location specified for the 'Location of the COBOL TXTLIB for Application Link-Edits' field. Correct the problem using ELAVMSET and rerun the preparation REXX exec.

ELA00429E Error during LKED - return code is rc

Explanation: The CMS LKED command returned a non-zero return code.

User Response: Refer to the preceding LKED messages and link-edit listing file (file type LKEDEDIT) for LKED error messages. Correct the problem and rerun the preparation REXX exec.

ELA00430E ENV= does not equal VMCMS or VMBATCH

Explanation: The value of the ENV= parameter for the ELAPREP EXEC must be either VMCMS or VMBATCH. The generation process should have built the preparation REXX exec with the correct value for this parameter.

User Response: Generate the program again to ensure that the ENV= parameter in the preparation REXX exec is not modified. Then rerun the preparation REXX exec. If the problem persists, contact the IBM Support Center.

ELA00431E PIPE RC = rc when trying to read LKED template file: filename LINKCTL filemode

Explanation: A CMS PIPE command was issued and generated a non-zero return code. If ELAPREP issued this message, the PIPE command was issued to read the specified file so that ELAPREP could process that file.

User Response: Refer to the online HELP information for the CMS PIPE command for information on the return code value. The file mode is the value that was specified for the ELAPREP VMFMODE= parameter. Ensure that the specified file exists. If necessary, generate the application again and ensure all the files associated with the application are successfully uploaded to the VM system. Then rerun the preparation REXX exec.

ELA00432E PIPE RC = rc when trying to create input file for LKED command

Explanation: A CMS PIPE command was issued and generated a non-zero return code. If ELAPREP issued this message, the PIPE command was issued to write a temporary file containing the LKED control statements. The file mode is the value that was specified for the ELAPREP VMFMODE= parameter.

User Response: Refer to the online HELP for the CMS PIPE command for information on the

return code value. Correct the problem and rerun the preparation REXX exec.

ELA00433E Error performing *command_name* command. Return code = *rc*.
Command was: *command_string*

Explanation: The indicated command generated a non-zero return code.

User Response: Refer to the VM CMS or DB2/VSE description of the command for information on the return code value. Correct the problem and rerun the preparation REXX exec.

ELA00434E Value *parm_value* for parameter *parm_name*= failed validation. It must be length 1 to 17

Explanation: The ELASQLAA EXEC detected an error in the value for the specified parameter.

User Response: Refer to the description of the parameter in the HELP screen for ELASQLAA (enter HELP ELA) or see the appendix on VM HELP files in *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information. Correct the indicated parameter and rerun the preparation REXX exec.

ELA00435E Value *parm_value* for parameter SQLSTMDE= failed validation. It must be either SINGLEUSER or MULTIUSER

Explanation: The ELASQLAA EXEC determined that an incorrect value was specified for the SQLSTMDE= parameter.

User Response: Refer to the description of the parameter in the HELP screen for ELASQLAA (enter HELP ELA) or see the appendix on VM HELP files in *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information. The user defined symbolic parameter SQLSTMDE can be used at generation time to set the SQLSTMDE= in the preparation REXX exec. Correct the indicated parameter and rerun the preparation REXX exec.

ELA00436E User ID *userid* part of SQLUSRPW= failed validation. It must be length 1 to 8

Explanation: The ELASQLAA EXEC detected an error in the value for the specified parameter.

User Response: Refer to the description of the parameter in the HELP screen for ELASQLAA (enter HELP ELA) or see the appendix on VM HELP files in *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information. The user defined symbolic parameter SQLUSRPW can be used at generation time to set the SQLUSRPW= in the preparation REXX exec. Correct the indicated parameter and rerun the preparation REXX exec.

ELA00437E An error occurred attempting to allocate *dataset_name*. Command *command* received a return code of *retcode*

Explanation: The ELACALLC EXEC was called to allocate the specified dataset. An error occurred issuing the CMS FILEDEF or DLBL command.

User Response: Ensure that the parameters passed to ELACALLC from the runtime exec are correct. Refer to the description of the parameters in the HELP screen for ELACALLC (enter HELP ELA) or see the appendix on VM HELP files in *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information. Correct the specification of the ELACALLC invocation and rerun the runtime exec.

ELA00439E File *dataset_name* expected for input and not found

Explanation: A CMS ESTATE command was issued and generated with a non-zero return code. If ELACALLC issued this message, it was checking the existence of the input dataset (ELACALLC *dsn* parameter).

User Response: Refer to the online HELP information for the CMS ESTATE command for information on possible error conditions from ESTATE. Ensure that the input file exists on the

file mode specified on the ELACALLC invocation. Correct the problem and rerun the runtime exec.

ELA00440E Application *appl_name* completed unsuccessfully with return code = *rc*

Explanation: The application runtime exec called the ELARUN EXEC to invoke the application. The ELARUN EXEC generated the specified non-zero return code.

User Response: Refer to the preceding error messages to determine the cause of this error.

ELA00441E VMLINK for *name = location* failed with RC = *rc*

Explanation: The CMS VMLINK command was issued to access a location with the following parameters:

name The installation variable name or the input parameter name

location A VMLINK nickname, CMS minidisk, or SFS directory name

rc The return code from the CMS VMLINK command

User Response: Refer to the online HELP information for the CMS VMLINK command for information on the return code value. Ensure that the location is specified correctly, that it exists, and that your user ID is authorized to access it.

ELA00442E *filename filetype* is not on an accessed disk

Explanation: The specified file is not on an accessed disk or SFS directory.

User Response: If ELARUN issued this message and was searching for the ELAVMDEF EXEC, ask the person who installed VisualAge Generator Server to verify that VisualAge Generator Server has been correctly installed.

ELA00443E ELAVMDEF EXEC was unable to set installation defaults

Explanation: The message text is self-explanatory.

User Response: Verify the existence of the installation defaults data file (ELACCDEF DATA) on the disk or SFS directory containing the executable VisualAge Generator Server files. This file was created by ELAVMSET during the installation of VisualAge Generator Server.

ELA00444E An error was found in the input to *routine_name*. Input supplied was: *input_parm*. The incorrect input parameter was *bad_parm*

Explanation: One of the parameters passed to the specified routine was incorrect.

User Response: Ensure that the parameters passed to the specified routine from the calling routine exec are correct. If ELACALLC was the routine that issued this message, refer to the description of the parameters in the HELP screen for ELACALLC (enter HELP ELA). Correct the specification of the ELACALLC invocation and rerun the runtime exec.

ELA00445E Valid file mode for DLBL for *dataset_name* could not be found

Explanation: ELACALLC EXEC attempted to find the first DOS disk in the CMS search order. If the file was being allocated for output, ELACALLC checks only disks accessed as R/W. There were no DOS disks (or R/W DOS disks) found in the CMS search order.

User Response: Link or access the appropriate DOS disks. You may want to include the LINK and ACCESS statements in the runtime exec.

ELA00446E {Null value was specified for AMODE= parameter | Value *parm_value* for parameter AMODE= failed validation}. It must be 24, 31 or ANY

Explanation: The value of the AMODE= parameter of ELAPREP must be 24, 31, or ANY.

User Response: Refer to the description of the parameter in the HELP screen for ELAPREP (enter HELP ELA). Correct the indicated parameter and rerun the preparation REXX exec.

ELA00447E {Null value was specified for RMODE= parameter | Value *parm_value* for parameter RMODE= failed validation}. It must be 24 or ANY

Explanation: The value of the RMODE= parameter of ELAPREP must be 24 or ANY.

User Response: Refer to the description of the parameter in the HELP screen for ELAPREP (enter HELP ELA). Correct the indicated parameter and rerun the preparation REXX exec.

ELA00448E Conflicting values were specified for the {SID= | SDB=} parameter. The conflicting values were: *parm1_value parm2_value*

Explanation: More than one unique value was specified for either the SID= or SDB= parameter during the execution of the application runtime REXX exec.

User Response: Edit the application runtime REXX execs for all of the applications executed in the run unit to verify that the SID= and SDB= values specified are the same. You can only specify one unique user ID, password, and database name using the SID= or SDB= parameters for the run unit.

ELA00449E This program requires CMS Level 11 or higher

Explanation: All VisualAge Generator Server programs require at least CMS Level 11 (VM/ESA Version 1 Release 2.2).

User Response: Re-IPL your virtual machine with the appropriate level of CMS or do not run this program. Contact your system programmer for assistance.

ELA00450E This program is not supported in a virtual machine running in 370 mode

Explanation: All VisualAge Generator Server programs require that the virtual machine be running in XA, XC or ESA mode.

User Response: Enter the CP SET MACHINE command with the appropriate parameters, and re-IPL your virtual machine. Contact your system programmer for assistance.

ELA00499E Undefined variable referenced on line *line_number* of filename *filetype* *filemode*

Explanation: The specified exec encountered an undefined variable referenced on the specified line. This condition indicates an error in the specified exec.

User Response: Contact your system programmer or IBM Software Support.

ELA03001I F3=EXIT F8=CONTINUE

Explanation: None.

User Response: None required.

ELA03002I F3=EXIT

Explanation: None.

User Response: None required.

ELA03003I CLEAR=EXIT

Explanation: None.

User Response: None required.

ELA03004I PF3=EXIT PF8=FORWARD

Explanation: None.

User Response: None required.

ELA03005I PF3=EXIT

Explanation: None.

User Response: None required.

ELA03006I PA1=CONTINUE

Explanation: None.

User Response: None required.

**ELA03007I VISUALGEN HOST SERVICES
VERSION 1.1**

Explanation: None.

User Response: None required.

**ELA03008I VISUALAGE GENERATOR
SERVER VERSION 3.0**

Explanation: None.

User Response: None required.

**FZE10014I ABEND %01C04 HAS
OCCURRED, TRAN= %02C04
%03C08 %04D05**

Explanation: CICS has detected an abend in the specified transaction. The time and date the abend was detected is listed. This message appears in the CSMT queue. If the abend is an ATNI, then the following information will also appear:

```
DATASTREAM FROM LAST TD QUEUE RECORD READ:  
data in hex format... *data in character format*  
DATASTREAM SENT TO THE DEVICE:  
data in hex format... *data in character format*
```

The above information shows the last transient data queue record read, as well as the data sent to the device which caused the ATNI abend. The data appears in both hex and character format, much like the data would appear in a CICS transaction dump.

User Response: The VisualAge Generator print transaction continues to run. Determine the cause of the CICS abend and run the transaction again if desired.

**FZE10040I PRINT TRANSACTION NOT
STARTED FROM TRANSIENT
DATA**

Explanation: The VisualAge Generator print transaction (program FZETPRT) received control for other than a transient data queue trigger level. Probable cause, XSPP entered at a terminal.

User Response: Contact your system administrator.

FZE10060P PARAMETER ERROR

Explanation: One or more of the input parameters was incorrectly specified.

User Response: If you were initializing a file, check the parameter list you specified. Correct it and try the procedure again. This message should not occur during the installation procedure. If this error occurs during installation, contact the IBM Support Center for assistance.

**FZE10061P ERROR OPENING %01C08 REG
15=%02X03, ERR=%03X03**

Explanation: An error occurred while attempting to open the named VSAM file.

User Response: Look up the return code in register 15 and the feedback (or reason) code in the appropriate VSAM manual for your operating environment. Correct the problem and try this procedure again.

FZE10062P ERROR WRITING %01C08 REG 15=%02X03, ERR=%03X03

Explanation: An error occurred while attempting to write to the specified VSAM file.

User Response: Look up the return code in register 15 and the feedback (or reason) code in the appropriate VSAM manual for your operating environment. Correct the problem and try this procedure again.

FZE10064I SUCCESSFUL COMPLETION

Explanation: This step in the installation procedure FZEZVCPO (MVS) finished correctly.

User Response: None required.

FZE10065I RECORDS READ: %01D08

Explanation: This shows the number of records read from the source statement library or from the system logical unit SYSIPT.

User Response: None required.

FZE10066I RECORDS WRITTEN: %01D08

Explanation: The indicated number of records were written to the VSAM output file.

User Response: None required.

FZE10067I FILE %01C08 ALREADY LOADED

Explanation: The specified output file has already been loaded or initialized. This message occurs when a file is being initialized or conditionally loaded.

User Response: None required.

FZE10068P SOURCE LIB I/O ERROR FOR FILE %01C08

Explanation: There was an error reading from the specified input file.

User Response: Check the listings for the return codes from the previous steps of the installation

procedure to determine if the source statement library installed correctly. If the return code was nonzero, correct the problem and run the previous step again. Then run this step again.

FZE10069P MISSING SOURCE MEMBER %01C08

Explanation: The specified source library member necessary for input to this step in the installation procedure is missing.

User Response: Check the listings for the return codes from the previous steps of the installation procedure to determine if the source statement library installed correctly. If the return code was nonzero, correct the problem and run the previous step again. Then run this step again.

MSG00001A Input file could not be opened

Explanation: The input file containing the messages could not be opened.

User Response: Do the following:

- Verify that the MSGFILE file has been allocated to the file that contains the existing messages.
- Verify that the MSGFILE file is a 256-byte relative VSAM file.

MSG00002A Output file could not be opened

Explanation: The external source format output file could not be opened.

User Response: Do the following:

- Verify that the MSGTBLE file has been allocated to the file where the external source format definitions will be stored.
- Verify that the file is serial fixed length 80.

MSG00003A Options file could not be opened

Explanation: The options file for the message file conversion utility, MSGOPTS, could not be opened.

User Response: Do the following:

- Verify that the MSGOPTS file has been allocated to the file that contains the conversion options.
- Verify that the file is serial fixed length 80.

MSG00004A Invalid or missing message table name

Explanation: The table name specified in the options file is invalid or missing. The NAME keyword must be specified and the value should:

- Contain six or seven characters.
- Contain a valid NLS code in the last three characters.
- The first character should be A-Z, or a-z. The remaining should be A-Z, a-z, or 0-9.

User Response: If the NAME keyword is not specified in the options file, include it. Otherwise, change the value to a valid value.

MSG00005I The message conversion completed successfully

Explanation: No errors were found during conversion. The external source format definition of the message table is in the MSGTBLE file.

User Response: None required.

MSG00006A Invalid keyword in options file

Explanation: An option specified in the options file (MSGOPTS) for the message conversion utility is invalid. Valid keywords are:

- NAME
- PREFIX
- APPLCHAR
- HIGHMSGN

User Response: Verify the following:

- The options file contains valid keywords.
- The keywords are spelled correctly.
- The options file contains no more than four keywords.
- The keywords have equal signs separating them from their values.

MSG00007A Invalid prefix value specified

Explanation: The PREFIX keyword was specified, but the value was not specified as either YES or NO.

User Response: Set the value of the PREFIX keyword to YES or NO.

MSG00008A Invalid program identifier character specified

Explanation: The APPLCHAR keyword was specified but the value was invalid. Valid values are:

- One character long
- A-Z, 0-9, #, \$, @

User Response: Specify a valid character value for the APPLCHAR keyword.

MSG00009A An error occurred while reading from the input file

Explanation: An error occurred when the Message File Conversion Utility attempted to read a record from the file MSGFILE.

User Response: Do the following:

- Verify that the proper access to file MSGFILE was provided to the Message File Conversion Utility.
- Verify that file MSGFILE is allocated to a VSAM user message file.

MSG00010A An error occurred while writing to the output file

Explanation: An error occurred when the Message File Conversion Utility attempted to write a record to file MSGTBLE.

User Response: Do the following:

- Verify that proper access to file MSGTBLE was provided to the Message File Conversion Utility.
- Verify that file MSGTBLE is allocated to a serial dataset with a fixed length of 80.

MSG00011A An error occurred while reading from the options file

Explanation: An error occurred when the Message File Conversion Utility attempted to read a record from file MSGOPTS.

User Response: Do the following:

- Verify that proper access to file MSGOPTS was provided to the Message File Conversion Utility.
- Verify that file MSGOPTS is allocated to a serial dataset with a fixed length of 80.

MSG00012P Value for highest message number invalid

Explanation: The value specified for the highest message number is not a number between 1 and 9999.

User Response: Do one of the following:

- Specify a number from 1 to 9999 on the HIGHMSGN keyword for better performance.
- Let the utility default the highest message number to 9999 by removing the HIGHMSGN keyword from the options file.

PRM00001P Invalid parameter group name %01C08

Explanation: The parameter group name specified is not valid. Parameter group names may be 1 through 8 alphanumeric characters.

User Response: Correct the parameter group name and reenter the request.

PRM00002I New parameter group being defined

Explanation: You have entered a parameter group name which has not been previously defined.

You may enter the parameters for the new parameter group to complete this definition. If you do not enter any parameters and you press Enter to save the group, then an empty group will be created.

User Response: None required.

PRM00003P Invalid selection character

Explanation: You have entered a selection code which is not valid. Valid selection codes are:

- 'S' - select a parameter group for update
- 'D' - delete an existing parameter group

User Response: Correct the selection character and reenter the request.

PRM00004P Already at top or bottom of list

Explanation: You attempted to do one of two things:

- Scroll forward on the last screen of the list
- Scroll backward on the first screen of the list.

No scrolling occurred.

User Response: Do not attempt to scroll past the start or the end of the list.

PRM00005I Function key not supported

Explanation: You have used a function key that is not supported by the facility. The keys which are available are described in the top portion of the map.

User Response: Check the description of what functions are available, and use a different function key.

PRM00006I Specified parameter group(s) not found

Explanation: You have requested to view a list of parameter groups, and no parameter group exists for the search conditions you have specified.

If you entered a question mark (?) to view a list of all parameter groups, then your parameter group file is empty.

User Response: If you have made an error, then correct and retry.

PRM00007P Unexpected I/O error occurred, RC = %01C08

Explanation: You have attempted an operation against the parameter group file and an I/O error has occurred. The operation was not completed.

This error indicates some damage has occurred to the parameter group file. This error should be corrected before any further maintenance to your parameter groups is attempted.

User Response: Contact your Systems Programmer.

PRM00008P File is full, parameter group cannot be added

Explanation: You have attempted to add a parameter group to your parameter group file, which is full. The parameter group has not been added.

User Response: Review your existing parameter groups to determine if any of them can be deleted. Deleting existing parameter groups will make room for new groups that you want to add. If you are not able to delete any existing parameter groups, then the parameter group file will have to be redefined to allow more entries.

PRM00009I Operation(s) successfully completed

Explanation: You have successfully completed the operation requested. The possible operations are:

1. Addition of a new parameter group.
2. Modification of an existing parameter group.
3. Deletion of an existing parameter group.

User Response: None required.

PRM00010P Parameter group file EZEPRMG not found

Explanation: Either the name was specified incorrectly or the file is not properly defined to the system.

User Response: Make sure the parameter group file is defined and associated with EZEPRMG as the DD name on MVS/TSO systems, or the FCT entry name on MVS CICS systems.

PRM00011P Unable to connect to parameter group file EZEPRMG

Explanation: The Parameter Group Utility was unable to connect to the parameter group file. The file must be associated and defined to the system.

User Response: Verify the file name specified has been defined and associated with EZEPRMG as the DD name on MVS/TSO, or the FCT entry name on MVS CICS systems.

Appendix. Understanding Syntax Diagrams

The syntax diagrams used throughout the documentation conform to the following conventions:

- The keywords can be listed in any order.

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

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

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

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

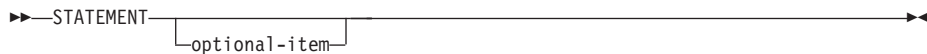
A syntax diagram that does not show the complete statement starts with the \blacktriangleright — symbol and ends with the — \blacktriangleright symbol. The \gg — symbol indicates the beginning of a statement.

A syntax diagram that show the complete statement starts with the \blacktriangleright — symbol and ends with the — \blacktriangleleft symbol.

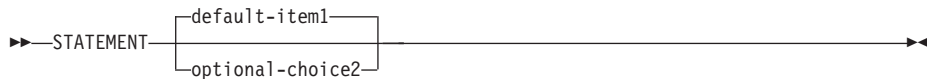
- Required items appear on the horizontal line (the main path).



- Optional items appear below the main path.



- Items positioned above the syntax diagram line are default parameters.



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

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



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



- An arrow returning to the left above the item indicates an item that you can repeat. Required items appear on the main line and optional items appear below the main line.



A repeat arrow indicates that you can make more than one choice from the grouped items, or repeat a single item.

- Keywords appear in uppercase (for example, PARM1). However, they can be uppercase or lowercase when they are entered. They must be spelled exactly as shown. Variables and acceptable values appear in all lowercase letters (for example, parmX). They represent names or values that you supply.
- If punctuation marks, parentheses, arithmetic operators, or other symbols are shown, you must enter them as part of the syntax.

Index

Special Characters

/FORMAT command 170
/HOLD command 171
/MODIFY command 169
/RUNFILE generation option 357
/SPA generation option 90, 167
/SYMPARM generation option 131

Numerics

16MB line, storage requests 299
1S78I error message 427
24-bit mode 299
31-bit mode 299

A

abend
ASPE 64, 312
codes
COBOL 282, 459, 584
IMS runtime 284
MVS CICS 274, 286
non-MVS CICS environments 276
preparation 245
system 279, 457, 581
VSE batch environments 454
VSE CICS 451
dumps
COBOL 228, 412, 550
VisualAge Generator Server 227, 411, 549
abend, VSE CICS
CICS
ASPE abend code 312
recovery considerations 323
abend codes
VisualAge Generator Server for MVS, VSE, and VM 579
activating trace sessions
IMS BMP 236
IMS/VS 232
MVS batch 236, 553
MVS CICS 232
MVS/TSO 236, 553
VSE batch 419
VSE CICS 416
adding
file name to the CICS file control table 317
job control statements 317
adding (*continued*)
the file name to the CICS file control table 70
the job control statements 69
addressing, extended 51
allocating files defined in resource association file 517
alternate index, defining 47, 293
alternate PCB, using 89
American National Standards control character 303, 305
American National Standards printer control character 55, 57, 58, 321
AMODE 15, 16
analyzing
CICS translator step 127, 345
compile step 127, 345, 491
DB2 precompile step 127, 491
DB2/VSE precompile step 344
detected errors 226, 410, 548
link-edit step 127, 345, 492
preparation listing 126, 344, 491
ANSISQL generation option 52, 300, 472
application,
load module storage for
VisualAge Generator Server for MVS, VSE, and VM 14
application plan for DB2 32
applying maintenance to
VisualAge Generator Server for MVS, VSE, and VM 3
ARI0415D error message 463
ARI0501I error message 247, 428
ARI0554E error message 247, 428
ARI0565I error message 247, 428
ARI5387E error message 247, 429
ARI5400I error message 247, 429
arithmetic overflow errors 209, 401
ASA (see American National Standards printer control character) 58, 321
ASPE abend 312
ASPE abend, preventing 64
attributes for DBCS, hardware 51, 297, 471
authority, granting to program users 129, 346, 493

B

backing up data 53, 302, 474
backup, maintaining copies of
production libraries 178, 371, 515
batch, message file conversion utility 195, 389
batch (see MVS batch) 83
batch (see VSE batch) 325, 362
batch print services program 119, 488
BIND
command 119
precompile messages 246
step, analyzing 128
bind command, data set 124
bind DB2 programs 52, 300, 472
buffer size, printing, MVS CICS 60
buffer size, printing, VSE CICS 308

C

CALL, transferring control in MVS CICS 64
CALL, transferring control in VSE CICS 312
catastrophic error 218, 547
cautions
empty KSDS data set, VSAM restriction 50, 296
PRTMPP parameter, line skip malfunction 61, 309
CEDA transaction, RDO 145
CEEUOPT program defaults module 130, 494
change or view, defaults - ELAC04 193, 387
change or view, options - ELAC02 190, 384
checking
access authorization 52, 301, 473
database authorization
IMS 96
MVS CICS 52, 301, 473
CICS, diagnostic control options 189, 383
CICS, ELAC transaction 189, 383
CICS, ELAM transaction 181, 375
CICS, ELAN transaction 183, 377
CICS, ELAU transaction 186, 380

CICS, parameter group, creating and maintaining 199, 393
 CICS, PRGM transaction 200, 394
 CICS, utilities, change diagnostic control options 190, 384
 CICS, utilities, default diagnostic control options 193, 387
 CICS, utilities, diagnostic control facility, ELAM 181, 375
 CICS, utilities, diagnostic control options, ELAC 189, 383
 CICS, utilities, diagnostic message printing 186, 380
 CICS, utilities, ELAC, diagnostic control options 189, 383
 CICS, utilities, ELAM, diagnostic control facility 181, 375
 CICS, utilities, ELAN, new copy utility 183, 377
 CICS, utilities, ELAU, diagnostic message printing 186, 380
 CICS, utilities, new copy utility, ELAN 183, 377
 CICS, utilities, parameter group utility, PRGM 200, 394
 CICS, utilities, PRGM, parameter group utility 200, 394
 CICS, utilities, view diagnostic control options 190, 384
 CICS/ESA, considerations 26
 CICS/ESA, monitoring and tuning 27
 CICS/MVS
 monitoring and tuning 27
 CICS transactions, passing transient data between 72, 320
 CICS utilities 181, 375
 CICS/SPARS (performance analysis reporting system) 69, 317
 clearing records from databases 98, 99
 client/server 71, 148, 319, 349, 354
 remote files 143
 client/server processing
 considerations 27
 client/server processing
 considerations, VSE CICS 35
 CLIST
 library data set 125
 modifying 135, 139, 497, 500
 runtime 139, 153, 500
 table 141
 templates 135, 498
 CMPAT parameter, IMS 89
 COBOL
 abend codes 282, 459, 584
 abend dumps 228, 412, 550
 abends under CICS 287
 availability 151, 511
 runtime messages 582
 status key values 271, 576
 COBOL, DATA compiler option 15, 16, 28
 COBOL, WSCLEAR option 41
 COBOL/370 runtime messages 282, 583
 COBOL compilation messages 561
 COBOL dynamic storage
 for VisualAge Generator Server
 for MVS, VSE, and VM 15
 codes
 abend
 COBOL 459
 abend, IMS 284
 return
 DL/I 447
 EZERT8 254, 431
 SQL 266, 445
 VisualAge Generator Server 256, 435
 codes, return, EZERT8 251
 commit point
 COMMIT 312
 EZECOMIT service routine 312
 IMS 91
 MVS batch 83
 MVS CICS 64
 MVS/TSO 79, 477
 VSE batch 327
 VSE CICS 312
 COMMIT service routine
 COMMIT, MVS batch 83
 COMMIT, MVS CICS 64
 common system return codes 565
 compatibility considerations, EZERT8 252, 565
 compile step, analyzing 127, 345, 491
 compiler options that affect performance 45, 291, 467
 compiling COBOL programs 521
 connection considerations 322, 328
 connection considerations for SQL programs 479
 considerations
 batch
 DB2/VSE 328, 330
 DL/I 323, 330
 considerations (*continued*)
 batch (*continued*)
 logical unit of work (LUW) 327
 program runtime support 330
 recovery 327
 system 325
 database integrity
 DB2, IMS 96
 DB2, MVS batch 84
 DB2, MVS CICS 74
 DB2, MVS/TSO 81, 482
 IMS 90
 MVS CICS 63
 database recovery
 IMS 90
 MVS CICS 63
 DB2
 MVS batch 84
 MVS CICS 73
 MVS/TSO 80, 479
 DB2 database recovery
 IMS 96
 MVS batch 84
 MVS CICS 74
 MVS/TSO 81
 DL/I
 IMS 97
 MVS batch 85
 MVS CICS 74
 MVS/TSO 81
 DL/I database integrity and recovery
 IMS 97
 MVS batch 85
 MVS CICS 75
 MVS/TSO 82
 message format services 105
 monitoring IMS 95
 MVS batch, system 83
 MVS/XA 51
 performance
 compiler options 45, 291, 467
 IMS 92, 94
 link pack area 94
 MVS CICS 65
 resident to CICS 292
 shared virtual area (SVA) 292
 VSE CICS 350, 351, 352, 356
 printing
 IMS 90
 MVS CICS 57
 MVS/TSO 78, 476

considerations (*continued*)
 recovery
 recovery, MVS batch 83
 recovery, MVS/TSO 79, 477
 residency
 IMS 92
 MVS CICS 65
 system
 backing up data 53, 302, 474
 DBCS 51, 297, 471
 extended addressing 51
 IMS 87
 MVS CICS 55
 tuning IMS 95
 VSE CICS
 DB2/VSE 322
 new copy 311
 performance 313
 printing 305
 recovery 312
 system 303
 considerations,
 client/server processing for MVS
 CICS 27
 client/server processing for VSE
 CICS 35
 DB2/VSE for VSE CICS 34
 VisualAge Generator Server for
 MVS, VSE, and VM 39
 VisualAge Generator Server for
 MVS, VSE, and VM, VSE batch
 installation 32
 control block 229, 551
 control character, American National
 Standards, printer 55, 57, 303, 305
 control region in IMS 31
 controlling
 diagnostic
 information in VSE CICS 356
 error reporting
 VSE batch 405
 VSE CICS 405
 controlling error reporting
 IMS 214
 MVS batch 215, 546
 MVS CICS 214
 conversational processing mode,
 MVS CICS 56
 conversion utility, message file 195,
 389, 533
 copybook 121
 copybook data set 125
 creating, runtime JCL 357
 creating MFS control blocks 168
 CREATX service routine, print
 destination 58, 306
 customizing
 see program directory 53, 302,
 474
 VisualAge Generator Server 53,
 302, 474
 customizing JCL procedures in
 VisualAge Generator Server for
 MVS, VSE, and VM 41
D
 DATA compiler option 15, 16, 28,
 29
 data file
 backing up 53, 302, 474
 defining 69, 317
 defining, program 47, 292
 program, defining 47, 292
 data queue
 extrapartition 320
 intrapartition 320
 transient 320
 data queue, transient 72
 data set
 bind command 124
 CLIST library 125
 DB2 database request module
 124
 DBRMLIB 124
 EZEBIND 124
 EZECLST 125
 EZECOPY 125
 EZEFOBJ 125
 EZEJCLP 125
 EZEJCLX 125
 EZEMFS 125
 EZEPCT 126
 EZEPCTL 125
 EZEPPT 126
 EZEPRINT data set 303, 320
 EZESRC 125
 load library 126
 map group format object module
 125
 message format services control
 block source 125
 message input descriptor layout
 copybook 125
 message output descriptor layout
 copybook 125
 MVS CICS
 PCT entries 126
 PPT entries 126
 object library 126
 parts directory 125
 destination 58, 306
 see program directory 53, 302,
 474
 VisualAge Generator Server 53,
 302, 474
 customizing JCL procedures in
 VisualAge Generator Server for
 MVS, VSE, and VM 41
D
 DATA compiler option 15, 16, 28,
 29
 data file
 backing up 53, 302, 474
 defining 69, 317
 defining, program 47, 292
 program, defining 47, 292
 data queue
 extrapartition 320
 intrapartition 320
 transient 320
 data queue, transient 72
 data set
 bind command 124
 CLIST library 125
 DB2 database request module
 124
 DBRMLIB 124
 EZEBIND 124
 EZECLST 125
 EZECOPY 125
 EZEFOBJ 125
 EZEJCLP 125
 EZEJCLX 125
 EZEMFS 125
 EZEPCT 126
 EZEPCTL 125
 EZEPPT 126
 EZEPRINT data set 303, 320
 EZESRC 125
 load library 126
 map group format object module
 125
 message format services control
 block source 125
 message input descriptor layout
 copybook 125
 message output descriptor layout
 copybook 125
 MVS CICS
 PCT entries 126
 PPT entries 126
 object library 126
 parts directory 125
 destination 58, 306
 see program directory 53, 302,
 474
 VisualAge Generator Server 53,
 302, 474
 customizing JCL procedures in
 VisualAge Generator Server for
 MVS, VSE, and VM 41
 data set (*continued*)
 SYSLIN 126
 SYSLMOD 126
 user 124
 work, temporary storage size
 315
 data set, loading KSDS files 50, 296
 database
 expanding 101
 integrity
 DL/I 324, 330
 SQL 323
 VSE CICS 311
 multiple, accessing 322
 multiple work 103
 recovery
 database integrity for DL/I
 324
 LUW (logical unit of work)
 324
 options 311
 relational database 323
 SQL 323, 330
 temporary storage size 315
 VSE CICS
 data integrity and recovery
 considerations 323
 partition 322
 work
 clearing records 98
 expanding 101
 maintaining 98
 temporary storage size 68
 database, work, temporary storage
 size 315
 database authorization
 checking
 IMS 96
 MVS 52, 301, 473
 database integrity and recovery
 considerations
 DB2
 IMS 96
 MVS batch 84
 MVS CICS 74
 MVS/TSO 81, 482
 DL/I
 IMS 97
 MVS batch 85
 MVS CICS 75
 MVS/TSO 82
 IMS 90
 MVS CICS 63
 database request module, DB2 124
 DB Tools product 95

- DB2
 - analyzing the precompile step 127, 491
 - checking authorization
 - IMS 96
 - MVS 52, 301, 473
 - considerations
 - MVS batch 84
 - MVS CICS 73
 - MVS/TSO 80, 479
 - database
 - request module data set 124
 - table space 101, 102
 - database integrity and recovery considerations
 - IMS 96
 - MVS batch 84
 - MVS CICS 74
 - MVS/TSO 81, 482
 - defining synonyms for tables 129, 493
 - granting authority to users 129, 493
 - precompile
 - messages 246
 - step, analyzing 127, 491
 - preparing and running programs 129, 493
 - programs
 - bind 52, 300, 472
 - RETAIN keyword, GRANT command 129, 493
 - work database
 - clearing records 99
 - expanding the table space 101
 - multiple 104
- DB2, application plan 32
- DB2, work database in IMS 32
- DB2 considerations
 - for MVS/TSO, VisualAge Generator Server for MVS, VSE, and VM 25
- DB2 considerations, for MVS CICS, VisualAge Generator Server for MVS, VSE, and VM 26
- DB2 considerations for IMS, VisualAge Generator Server for MVS, VSE, and VM 30
- DB2/VSE
 - analyzing the precompile step 344
 - defining synonyms for tables 347
 - granting authority to users 346
- DB2/VSE (*continued*)
 - precompile
 - messages 247, 428
 - step, analyzing 344
 - preparing and running programs 346
 - running
 - main program 363
 - main program with DL/I, VSE batch JCL 367
- DB2/VSE considerations, VSE CICS 34
- DB2/VSE considerations, VSE environment 33
- DB2/VSE precompiler messages 562
- DBCS (double-byte character set)
 - hardware attributes 51, 297, 471
 - MSGPOPT=CCTRL option 310
 - printer 72, 320
- DBCS considerations, printing, MVS CICS 62
- DBCS data on a non-DBCS terminal 171
- DBCS(double-byte character set)
 - printing, MVS CICS 58
- DBP value, DL/I 312
- DBRMLIB 124
- DCAPRMG file, parameter group for FZETPRT 59, 307
 - transient data queue 306
 - trigger level 306
- DCT (destination control table)
 - entries 147, 354
 - printing, DBCS 62
 - printing, DBCS (double-byte character set) 310, 321
 - sample entry 72
 - transient data queue 58, 306, 320
 - trigger level 58, 306
- DD statements by file type 136, 498
- deactivating a trace session 238, 422, 557
- deallocating files defined in resource association file 517
- default
 - message queue file, IMS MVS/TSO 78, 476
 - print destination, IMS 90
- defining
 - alternate index 47, 293
 - data files 69, 317
 - ESDS (serial) data set 47, 293
- defining (*continued*)
 - intrapartition transient data files 320
 - intrapartition transient data queues 320
 - KSDS (indexed) data set 47, 293
 - program data files 69, 317
 - program specification block, IMS 88
 - program specification block (PSB) 347
 - program VSAM data files
 - VSE batch 325
 - PSB (program specification block) 347
 - RRDS (relative) data set 47, 293
 - synonyms for DB2 tables 129, 493
 - synonyms for DB2/VSE tables 347
 - transient data queue 72
 - transient data queues 320
 - VSAM data files 47, 292, 325, 468
- defining, extrapartition transient data 73, 321
- defining, intrapartition transient data 72, 320
- DEFTEST AMSERV 469
- deleting old records from the work database 98
- descriptions
 - IMS files 87
 - MVS CICS files 55
- destination, default print, IMS 90
- destination control table (DCT)
 - entries 147, 354
 - printing, DBCS 62
 - printing, DBCS (double-byte character set) 310, 321
 - sample entry 72
 - transient data queue 58, 306, 320
 - trigger level 58, 306
- detecting errors 207, 399, 541
- determining position in program 230, 552
- DFHAC2016 messages 286
- DFHAC2206 messages 286
- DFHTCT TYPE=TERMINAL entry 29
- DFS057I error message 283
- DFS064 error message 283
- DFS182 error message 283
- DFS2082 error message 172, 284

DFS2766I error message 172, 284
DFS555I error message 172, 283
diagnosing problems 207, 399, 541
diagnostic control facility, change or view defaults 193, 387
diagnostic control facility, change or view options 190, 384
diagnostic control facility, CICS utilities 181, 375
diagnostic control facility, ELAC transaction 189, 383
diagnostic message print utility, ELAU 186, 380
disk storage requirements
 for VisualAge Generator Server for MVS, VSE, and VM 18
DISPLAY process option 57, 305
DL/I
 ACT
 (application control table), CICS programs, scheduling PSBs 323
 (application control table), MPS batch jobs 331
 application control table (ACT), CICS programs, scheduling PSBs 323
 table (ACT), MPS batch jobs 331
 considerations
 IMS 97
 MVS batch 85
 MVS CICS 74
 MVS/TSO 81
 integrity and recovery considerations
 IMS 97
 MVS batch 85
 MVS CICS 75
 MVS/TSO 82
 MPS batch job with DL/I databases, ACT entry 331
 program runtime support 81
 running
 main program 365
 main program with DB2/VSE, VSE batch JCL 367
 runtime support 330
 scheduling PSBs, ACT entry 323
 specifying DATA compiler option 15
 support in VSE batch programs 330
 work database
 clearing records 98
DL/I (continued)
 work database (continued)
 expanding the database 101
 multiple 104
DL/I considerations
 for MVS CICS in VisualAge Generator Server for MVS, VSE, and VM 26
 for MVS/TSO, VisualAge Generator Server for MVS, VSE, and VM 24
DL/I considerations, VSE batch 32
DL/I considerations, VSE CICS 34
DL/I status codes 268
DL/I work database in IMS 32
DLBL name
 file control table (FCT) entry 318
 VSE CICS startup JCL 318
DLZ109I message 463
double-byte character set (DBCS)
 hardware attributes 51, 297, 471
 MSGPOPT=CCONTRL option 310
 printer 72, 320
DSNH039I messages 246
DSNX100I messages 246
dump listing file, snap, IMS 87
dumps, analyzing different types of 549
DXFR statement, transferring control in CICS 64
DXFR statement, transferring control in VSE CICS 312
dynamic
 backout, VSE CICS 312
 interface plan 53
 linkage 128, 346
dynamic storage
 in VisualAge Generator Server for MVS, VSE, and VM 16
dynamic storage utilization, VisualAge Generator Server for MVS, VSE, and VM 16
E
EATTR generation parameter 108
ELA2SSQL module 93
ELA2SSQX module 93
ELA2SSQY module 93
ELAC, diagnostic control facility 189, 383
ELAC02 panel, change or view options 190, 384
ELAC04 panel, change or view defaults 193, 387
ELACALLC EXEC 517
ELACCDEF DATA control file 531
ELACJKWD member 104
ELADIAG file 87
ELAM, CICS utilities menu 181, 375
ELAN, new copy utility 183, 377
ELANCCcc module 93
ELAPCB macro 88, 89
ELAPREP EXEC 521
ELAPRINT system output file 78, 87, 476
ELARPRTM load module 93
ELARPRTR load module 93
ELARSDCB load module 93
ELARUN EXEC 527
ELASNAP file 87
ELASQLAA EXEC 528
ELAU 186, 380
ELAU, diagnostic message printing utility 186, 380
ELAVMSET EXEC 531
ELAWKJIC2 member 99
ELAWKJCD member 98
ELAWORK work database PCB 89
ELAWORK2 DL/I work database 104
emulating IBM 3270 devices 51, 297, 471
ERRDEST message queue 221
error
 arithmetic overflow 209, 401
 detection 207, 399, 541
 file
 definition 209, 401
 I/O 208, 400
 processing 208, 400
 logic 210, 402
 message
 file 87
 panel 220, 547
 program interaction 211, 403
 reporting 213, 546
 description 405
 IMS 214, 215
 in IMS 172
 MVS batch 215
 MVS/TSO 215
 summary 215, 406
 screen editing 207, 399
 system 213, 404
ESDS (serial) data set definition 47, 293
ESDS (serial) define cluster 47, 293
examples, runtime REXX exec 502

examples of runtime JCL, VSE batch programs 362

expanding
the table space (DB2) 101
work database 101

express alternate PCB 89

extended addressing
considerations 51
VisualAge Generator Server 51

external source format table,
converting a message file to 533

external work file, backing up 53, 302, 474

extrapartition transient data,
defining 73, 321

EZEBIND data set 124

EZECLST data set 125

EZECOMIT service routine 312
IMS 91
MVS batch 83
MVS CICS 64
MVS/TSO 80, 477

EZECONCT service routine 329

EZECOPY data set 125

EZEDESTP special function word
77, 306, 475

EZEDLSTC special function word
82

EZEFOBJ data set 125

EZEJCLP data set 125

EZEJCLX data set 125

EZEMFS data set 125

EZEOPBMP generation option file
105

EZEOPIMS generation option file
105

EZEORDER special function word
210, 402

EZEP transaction 55, 58, 72, 306, 320

EZEP transient data queue 303

EZEPCT data set 126

EZEPCTL data set 125

EZEPPT data set 126

EZEPRINT data set 303, 320
IMS 90
MVS CICS 55
MVS/TSO 77, 78, 475, 476
specify as PRO1 72, 320

EZEPRMG file
MVS CICS 55
parameter group for FZETPRT
59

EZEPRMG file, parameter group for
FZETPRT 303, 307

EZEROLLB service routine 312, 327
IMS 91
MVS batch 84
MVS CICS 64
MVS/TSO 80, 478

EZERT8 special function word
compatibility considerations 252
, 565
return codes 254, 566

EZERT8 special function word,
return codes 251, 431

EZESEGM special function word
56, 304

EZESRC data set 125

EZETRACE data set 78, 476

EZEZ transaction 58, 72, 306, 320

F

FCT (file control table)
data set name, user 318
definition errors 401
description 303
DLBL name and CICS startup
JCL 318
entries 147, 354
FILE name parameter 318
I/O errors 400
parameter group 303
processing errors 400
user data file 70
VSE CICS 303

file
control table 147
default message queue, IMS 87
definition errors 209
description
IMS 87
MVS CICS 55
descriptions 55
error message 87
I/O errors 208
parameter group 55
processing errors 208
snap dump listing, IMS 87
system output 78
trace 78
utilities print 55

file allocation considerations 500

file control table (FCT)
data set name, user 318
definition errors 401
description 303
DLBL name and CICS startup
JCL 318
entries 147, 354
FILE name parameter 318

file control table (FCT) *(continued)*
I/O errors 400
parameter group 303
processing errors 400
user data file 70
VSE CICS 303

files from generation 114, 485

fold tables 53, 302, 474

form-feed, printing 58, 306

form-feed order (see American
National Standards printer control
character) 58

FORMFD parameter
option=NO, forms alignment 58
, 306
parameter group for EZEP or
EZEZ 62, 310
used with FZETPRT program 59
using with FZETPRT program
307

function
new copy 63
preload, IMS 92

function, new copy 311

FZEREBLK utility
initializing indexed files 296

FZETPRT program 62
DBCS considerations 62
EZEP or EZEZ transaction 72,
320
print program 310
special parameter group 59, 307
terminal printing support in MVS
CICS 58
terminal printing support in VSE
CICS 306
VSE CICS table entries, DBCS
only 310

FZEZREBO utility, initializing
indexed files 50, 296

G

generated applications, making
available for MVS/TSO 25

generated applications, with PL/I
programs 41

generated modules availability 151,
511

generating
file allocation statements,
templates for 498

generating, application control block
88

generation
and compiler options that affect
performance 45, 291, 467

generation (*continued*)
 linkage editor control (LED) file 346
 options
 /ANSISQL 52, 300, 472
 /COMMENTLEVEL 227, 411, 549
 /ERRDEST 214, 221, 224
 /EZETRACE 231, 553
 /FASTPATH 214, 218, 283, 284
 /INITADDWS 279, 457, 581
 /LOG 214, 222
 /MFSDEV 107, 283, 284
 /MFSEATTR 106, 283, 284
 /MFSEATTRNCD 106
 /MFSIGNORE 107, 283, 284
 /MFSTEST 283
 /NLS 280, 458, 582
 /NOFASTPATH 214, 218
 /NOLOG 214
 /NOMFSEATTR 106
 /NOMFSIGNORE 107
 /NOMFSTEST 283
 /NONUMOVFL 210, 402
 /NORECOVERY 214
 /NUMOVFL 210, 402
 /RECOVERY 214, 217
 /SPA 90, 167
 /SYMPARM 131
 /TRACE 415
 language code 221, 223, 225, 409
 performance considerations 45, 291, 467
 options, /RUNFILE 357
 options, /TARGNLS 409
 output files 114, 485
 output of 335
 parameter
 EATTR 108
 NOEATTR 108
 generation, options, /INITRECD 279, 457, 581
 granting authority to program users 129, 346, 493

H

hardware attributes for DBCS 51, 297, 471
 hardware requirements
 VisualAge Generator Server for MVS, VSE, and VM 5
 HELP files, online 517

I
 IBM 3270 device, emulating 51, 297, 471
 IBM 5550 family of terminals 51, 297, 471
 IDCAMS program
 BLDINDEX command 47, 293
 DEFINE PATH command 47, 293
 loading indexed files 50
 REPRO command 47, 50, 293, 296
 IESIO230I error message 427
 IGYGR1471W message 429
 IGYOP3091W error message 248, 429, 561
 IGYOP3093W error message 248, 429
 IGYOP3094W error message 248, 429
 IGYPA3013W error message 248, 430
 IGYPG3113W error message 248, 430
 IGYPS2015I error message 248, 430
 IGYPS2023I error message 248, 430
 IGYSC2025W error message 249, 430
 IGZ prefix messages 582
 IGZ033S error message 282
 IGZ064S error message 282
 IGZ066S error message 282
 IGZ075S error message 282
 improving
 performance 95
 library lookaside (LLA) 46
 link pack area (LPA) 46
 virtual lookaside facility (VLF) 46
 response time 94
 IMS
 commit point 91
 database
 authorization checking 96
 integrity considerations 90
 recovery considerations, DB2 96
 recovery considerations, DL/I 90
 default
 message queue file 87
 print destination 90
 DL/I considerations 97
 ELAPCB macro 89

IMS (*continued*)
 error
 controlling, generation options 214
 messages 172
 reporting 172
 file descriptions 87
 HIPERSPACE buffer usage 94
 installation considerations 29
 integrity considerations, DB2 96
 log format 222
 logical unit of work 97
 monitoring 95
 new modules 169
 performance considerations 94
 preload function 92
 preloading
 program modules 94
 VisualAge Generator Server modules 93
 preparation 165
 processing modes 90
 program specification block, defining 88
 residency considerations 92
 rollback 91
 runtime
 abend codes 284
 messages 283
 segmented mode 90
 single-segmented mode 90
 snap dump listing file 87
 system considerations 87
 system definitions
 batch-oriented BMP program 167
 batch program as an MPP 167
 interactive program 166
 parameters 165
 transaction-oriented BMP 168
 system printing considerations 90
 IMS, control region 31
 IMS, DB2 considerations in VisualAge Generator Server for MVS, VSE, and VM 30
 IMS, DB2 work database considerations 32
 IMS, DL/I work database considerations 32
 IMS, monitoring and tuning 30

- IMS, security considerations for
 - VisualAge Generator Server for MVS, VSE, and VM 30
 - IMS, system definition 31
 - IMS/BMP, activating trace sessions 236
 - IMS DC monitor facilities 30
 - IMS/ESA exploitation 29
 - IMS Performance Analysis and Reporting System 30
 - IMS/VS, message format service (MFS) Control Blocks 95
 - IMSPARS 30, 95
 - indexed (KSDS) data set
 - define cluster 47, 293
 - loading 50, 296
 - INQUIRESET programs 28
 - installation,
 - VisualAge Generator Server for MVS, VSE, and VM, preparing for 3
 - installation considerations for VSE batch 32
 - installation program 531
 - integrity considerations, database DB2
 - IMS 96
 - MVS batch 84
 - MVS CICS 74
 - MVS/TSO 81, 482
 - DL/I
 - IMS 97
 - MVS batch 85
 - MVS CICS 75
 - MVS/TSO 82
 - IMS 90
 - integrity considerations, database, SQL, batch 330
 - intrapartition transient data, defining 72, 320
 - intrapartition transient data files 320
 - invoking the SQL preprocessor for VisualAge Generator COBOL programs 528
 - INW0092I error message 427
- J**
- JCL
 - by environment 136, 498
 - examples of runtime 362
 - modifying 135, 137, 497
 - modifying runtime 359
 - tailoring before generation 358
 - templates 135, 136, 340, 498
- job stream data set
 - preparation 125
 - runtime 125
- K**
- KSDS (indexed) define cluster 47, 293
- L**
- LE, runtime messages 281, 458
 - LE runtime options 130, 494
 - LED (linkage editor control) file, generation 346
 - libraries, maintaining copies of
 - production for backup 178, 371, 515
 - library 53, 302, 474
 - library lookaside (LLA), MVS/ESA 46
 - line character format 57, 305
 - link-edit step
 - analyzing 127, 492
 - CEEUOPT module selection 131, 496
 - LE runtime options 131, 495
 - link-edit step, analyzing 345
 - link pack area
 - loading 94
 - performance considerations 94
 - LINK statement, transferring control in CICS 64
 - LINK statement, transferring control in VSE CICS 312
 - linkage editor control (LED) file, generation 346
 - linking compiled COBOL programs and generated object modules 521
 - listing file 340, 489
 - listing file, snap dump, IMS 87
 - load library data set 126
 - load module 127, 492
 - preloading 93
 - load module storage, VisualAge Generator Server for MVS, VSE, and VM application 14
 - load modules storage, VisualAge Generator Server for MVS, VSE, and VM 13
 - loading
 - an object, VSE CICS performance considerations 313
 - modules into link pack area 94
 - loading data into a VSAM file 470
 - logic errors 210, 402
 - logical unit of work (LUW)
 - IMS 96, 97
- logical unit of work (LUW)
 - (continued)
 - MVS batch 83
 - MVS CICS 64
 - MVS/TSO 79, 477
 - logical unit of work (LUW), description 323, 324, 330
 - LUW (logical unit of work), description 323, 324, 330
- M**
- macro, ELAPCB 89
 - maintaining backup copies of
 - production libraries 178, 371, 515
 - maintaining work database 98
 - maintenance, applying to
 - VisualAge Generator Server for MVS, VSE, and VM 3
 - making
 - new modules available in VSE CICS 355
 - VisualAge Generator Server COBOL and the generated modules available 357
 - making modules available, MVS/TSO and VisualAge Generator Server for MVS, VSE, and VM 25
 - managing VSAM data files 468
 - map group
 - format module 120, 488
 - profile block 229, 551
 - map group, profile block 413
 - map group format module data set 125
 - message
 - format services
 - considerations 105
 - control block data set 125
 - description 51, 105
 - input descriptor layout copybook data set 125
 - output descriptor layout copybook data set 125
 - queue file, default, IMS 87
 - message, 1S78I 427
 - message, IESI0230I 427
 - message file conversion utility 195, 389, 533
 - message format service (MFS) Control Blocks in IMS 95
 - messages
 - ARI0415D 463
 - ARI0501I 247, 428
 - ARI0554E 247, 428
 - ARI0565I 247, 428

messages (*continued*)
 ARI5387E 247, 429
 ARI5400I 247, 429
 COBOL 582
 COBOL/370 runtime 282, 583
 DB2/VSE precompiler 562
 DFHAC2016 286
 DFHAC2206 286
 DFS057I 283
 DFS064 283
 DFS182 283
 DFS2082 172, 284
 DFS2766I 172, 284
 DFS555I 172, 283
 DLZ109I 463
 DSNH039I 246
 DSNX100I 246
 IGYGR1471W 429
 IGYOP3091W 248, 429, 561
 IGYOP3093W 248, 429
 IGYOP3094W 248, 429
 IGYPA3013W 248, 430
 IGYPG3113W 248, 430
 IGYPS2015I 248, 430
 IGYPS2023I 248, 430
 IGZ0033I 582
 IGZ0033S 282
 IGZ0064S 282
 IGZ0066S 282
 IGZ0075S 282
 IMS runtime 283
 MFS generation 245
 MVS runtime 285
 preparation 245, 427, 561
 runtime
 COBOL 582
 COBOL/370 282, 583
 IMS 283
 MVS 285
 VS COBOL II 582
 messages, IGYSC2025W 249, 430
 messages, INW0092I 427
 messages, NW0002I 428
 MFS
 control blocks 120, 168
 generation messages 245
 MID/MOD layout 121
 print services program 119
 MFSEDEV generation option 107
 MFSEATTR generation option 106
 MFSEATTRNCD generation option 106
 MFSIGNORE generation option 107
 mode
 MVS CICS execution,
 performance considerations 67
 processing
 IMS 90
 MVS CICS 56
 VSE CICS processing 304
 performance considerations 315
 types 304
 models
 CLIST 141
 JCL 136
 modifying
 CICS resource tables 349
 CICS startup JCL 355
 IMS system definition parameters 165
 JCL or CLISTs 135
 runtime
 CLISTs 139, 497
 JCL 137
 runtime JCL 357, 359
 modifying MVS CICS resource tables 28
 modifying VSE CICS resource tables 36
 module availability 477
 modules
 IMS 169
 loading into link pack area 94
 MVS CICS 148
 preloading 94
 modules in memory 46
 monitoring
 and tuning, VSE CICS 317
 IMS system 95
 performance 95
 performance in VSE CICS 317
 monitoring and tuning, IMS, VisualAge Generator Server for MVS, VSE, and VM 30
 monitoring and tuning, MVS CICS 27
 monitoring and tuning, VSE CICS 35
 moving
 prepared programs
 to another VSE system 371
 to other systems 371
 moving prepared programs 177, 515
 MSLs
 CICS/ESA considerations for VisualAge Generator Server for MVS, VSE, and VM 26
 client/server processing considerations 27
 DB2 considerations for VisualAge Generator Server for MVS, VSE, and VM 25
 DL/I considerations 24
 DL/I considerations for VisualAge Generator Server for MVS, VSE, and VM 24
 installation considerations 24, 26
 installation considerations, VisualAge Generator Server for MVS, VSE, and VM 24, 37
 making generated programs available for VisualAge Generator Server for MVS, VSE, and VM 25
 making modules available for VisualAge Generator Server for MVS, VSE, and VM 25
 monitoring and tuning 27
 MVS CICS 28
 program control table 28
 program processing table 28
 resource control table 28
 security considerations 27
 security considerations for VisualAge Generator Server for MVS, VSE, and VM 25
 spool files 29
 system initialization table 28
 temporary storage queues for VisualAge Generator Server for MVS, VSE, and VM 29
 multiple printers, printing 309
 multiple work databases 103
 MVS
 preparation 159
 runtime messages 285
 VisualAge Generator Server for MVS, VSE, and VM installation 3
 MVS, VisualAge Generator Server, software requirements 7
 MVS batch
 activating trace sessions 236, 553
 commit point 83, 477
 DB2 considerations 84
 DL/I considerations 85

- MVS batch (*continued*)
 - error reporting, controlling 215, 546
 - logical unit of work 83, 477
 - message file conversion utility 533
 - recovery considerations 83
 - rollback 84, 477
- MVS CICS
 - abend codes 286
 - activating trace sessions 232
 - CALL statement 64
 - CICSPARS (performance analysis reporting system) 69
 - commit point 64
 - database
 - integrity considerations 63
 - recovery considerations 63, 74
 - DB2 considerations 73
 - destination control table (DCT)
 - printing, DBCS 62
 - sample entry 72
 - transient data queue name 58
 - DL/I considerations 74
 - DXFR statement 64
 - EZEP transaction 58
 - EZEZ transaction 58
 - file descriptions 55
 - LINK statement 64
 - logical unit of work 64
 - (LUW), ending 64
 - LUW (logical unit of work), ending 64
 - mode, pseudoconversational, residency consideration 67
 - new modules 148
 - parameter group
 - print file 55
 - PCT (program control table), printing, DBCS 62
 - performance
 - considerations 65
 - monitoring 69
 - performance analysis reporting system (CICSPARS) 69
 - preparation 143
 - print destination, specifying in DCT 72
 - printing
 - American National Standards control character 57
 - buffer size 60
- MVS CICS (*continued*)
 - printing (*continued*)
 - DBCS (double-byte character set) 58, 62
 - DCT (destination control table) 58
 - destination control table (DCT) 58, 72
 - double-byte character set (DBCS) 58
 - enqueueing on transient data queue 57
 - EZEP transaction 58
 - EZEPRINT data set 72
 - EZEZ transaction 58
 - file description 55
 - form-feed 58
 - FORMFD=NO parameter 58
 - FZETPRT program 59, 72
 - line character format 57
 - parameter, PRITYP 61
 - PCT (program control table), FZETPRT program 62
 - program control table (PCT) 62, 72
 - PRTBUF parameter 59
 - PRTMPP parameter 59
 - PRITYP parameter 59
 - SEND command 60
 - temporary storage queue 57
 - terminal control table (TCT), entry 72
 - transient data queue 57, 72
 - processing mode
 - performance 68
 - types 56
 - program control table (PCT)
 - DTB=YES and DBP value 72
 - printing, DBCS 62
 - pseudoconversational
 - processing mode 67
 - programs and residency 67
 - residency
 - considerations 65, 67
 - general rules 66
 - resource tables 143
 - rollback 64
 - startup JCL 148
 - system considerations 55
 - tables, recovery options 64
 - temporary storage, queue 57
 - terminal control table (TCT), entry 72
 - terminal printing 58
- MVS CICS (*continued*)
 - transaction
 - EZEP 58
 - EZEZ 58
 - PR01 transient data queue 72
 - transferring control
 - CALL statement 64
 - DXFR statement 64
 - LINK statement 64
 - XCTL statement 64
 - transient data queue 58
 - translator step 127
 - XCTL 64
- MVS CICS, DB2 considerations in VisualAge Generator Server for MVS, VSE, and VM 26
- MVS CICS, DL/I considerations 26
- MVS CICS, utilities, VisualAge Generator Server for MVS, VSE, and VM 27
- MVS CICS storage facilities used by VisualAge Generator Server for MVS, VSE, and VM 17
- MVS/TSO
 - activating trace sessions 236, 553
 - CLIST 153
 - CLIST library data set 125
 - considerations, DB2 80, 479
 - database
 - integrity considerations, DB2 81, 482
 - recovery considerations, DB2 81, 482
 - DL/I considerations 81
 - error reporting, controlling 215, 546
 - logical unit of work 79, 477
 - preparation 151, 511
 - print destination 78, 476
 - printing considerations 78, 476
 - recovery considerations 79, 477
 - rollback 80, 478
- MVS/XA considerations 51

N

- new copy
 - function 311
 - utility 311
- new copy, function 63
- new copy utility 183, 377
- new copy utility, ELAN 183, 377
- new-line order, printing, VSE CICS 306

new modules
 IMS 169
 MVS CICS 148
 NOEATTR generation parameter 108
 NOMFSEATTR generation option 106
 NOMFSIGNORE generation option 106
 nonsegmented processing mode, MVS CICS 56
 NW0002I error message 428

O

object library data set 126

objects
 generated for
 map groups 339
 tables 339
 produced by the generation facility 336

objects generated
 application COBOL program 117, 487
 batch print services program 119, 488
 BIND command 119
 from generation 114, 485
 map group format module 120, 488
 MFS
 control blocks 120
 MID/MOD layout 121
 print services program 119
 online print services program 119, 488
 parts control file 121, 489
 preparation JCL 121, 489
 runtime
 CLIST 118, 488
 JCL 118
 table program 119, 488
 online HELP files 517
 online print services program 119, 488

option
 generation
 /MFSDEV 107
 /MFSEATTR 106
 /MFSEATTRNCD 106
 /MFSIGNORE 107
 /NOMFSEATTR 106
 /NOMFSIGNORE 107
 LE runtime 130
 preloading
 program modules, IMS 94

option (*continued*)
 preloading (*continued*)
 VisualAge Generator Server modules, IMS 93
 recovery 64
 SPA 90

optional files 476

output of generation 114, 485
 output of program generation 335

P

page alignment, printing, VSE CICS 306

panels, Parameter Group Definition (PRGM02) 202, 396

panels, Parameter Group List Display (PRGM01) 201, 395

panels, Parameter Group Specification (PRGM00) 200, 394

parameter
 CMPAT, IMS 89
 FILE name, FCT (file control table) entry 318
 generation
 EATTR 108
 NOEATTR 108
 group associated with FZETPRT program
 DCAPRMG file 59, 307
 EZEPRMG file 59, 307
 resident 66, 314
 parameter, generation, NOEATTR 108

Parameter Group Definition panel (PRGM02) 202, 396

parameter group file, EZEPRMG data set, MVS CICS 55

parameter group file, EZEPRMG data set, VSE CICS 303

Parameter Group List Display panel (PRGM01) 201, 395

Parameter Group Specification panel (PRGM00) 200, 394

parameters, WORK in ELAPCB 89

parts control file 121, 341, 489

parts directory data set 125

passing, transient data between CICS transactions 72, 320

passing transient data between VSE CICS transactions 320

PCT (program control table)
 DTB=YES and DBP value 312
 entries 145, 349, 351
 FZETPRT program 62, 310, 312, 320
 generation output 117

PCT (program control table) (*continued*)
 interaction errors 403
 specification block (PSB), defining 347

performance
 analysis reporting system (CICSPARS) 69, 317
 considerations 45, 291, 467
 general 46
 IMS 92, 95
 IMS/ESA 94, 95
 MVS batch 86
 MVS CICS 65
 VSE batch 331
 VSE CICS 356

conversational mode, VSE CICS 316

conversational processing, MVS CICS 68

database temporary storage size 315

DTB=NO, PCT entry 352

generation and compiler options 45, 291, 467

HIPERSPACE buffers for IMS 94

library lookaside (LLA) 46

limiting MFS control blocks 95

link pack area 46

making programs resident 356

monitoring
 IMS 95
 MVS CICS 69

monitoring, VSE CICS 317

monitoring and tuning the IMS system 95

phases in memory 292

preload modules 169

processing mode in VSE CICS 316

PRTBUF parameter, terminal printing 69, 316

pseudoconversational mode, VSE CICS 316

pseudoconversational processing, MVS CICS 68

RES(YES), DEF PROG entry 351

RES=YES, DFHPPT entry 350, 351

RES(YES) parameter, RDO
 DEFINE PROGRAM command 145

RES=YES parameter, PPT entry, CICS table 144

residency considerations 313

- performance (*continued*)
 - shared virtual area (SVA) 292
 - terminal printing, MVS CICS 69
 - terminal printing, VSE CICS 316
 - tuning IMS 95
 - virtual lookaside facility (VLF) 46
- performance, monitoring and tuning IMS 30
- Performance Analysis and Reporting System (PARS) 95
- performance considerations
 - VisualAge Generator Server for MVS, VSE, and VM 40
- phases in memory 292
- PL/I programs 41
- plan, DB2 53
- PPT (processing program table)
 - defining programs to CICS 113
 - entries 144, 349
 - generation output 117
 - RES=YES parameter 144
 - resident parameter 314
 - samples 144
 - TYPE=GROUP and FN=BACKOUT entries 64
- PR01 transient data queue 72, 320
- precompile messages
 - BIND 246
 - DB2 246
- precompile messages, DB2/VSE 247, 428
- preload function, IMS 92
- preloading
 - objects, IMS 92
 - print services
 - description 169
 - module 93
 - program 94
 - program 169
 - program modules 93, 94
 - service module 93
 - table modules 93, 169
 - VisualAge Generator Server modules, IMS 93
- preparation
 - abend codes 245
 - command file 340
 - JCL 121, 126, 343, 489, 491
 - DB2/VSE 340
 - for VSE environments 340
 - map group format modules 340
 - VSE CICS 340
 - listing 126, 344, 491
- preparation (*continued*)
 - messages 245, 427, 561
 - preparation command file 489
 - preparation job stream data set 125
 - PREPARE subcommand 126, 343, 491
 - preparing
 - and running
 - DB2/VSE programs 346
 - generated programs in VSE CICS 349
 - generated programs 123, 343, 491
 - VSE batch 357
 - programs
 - for VSE batch 357
 - for VSE CICS 349
 - running the preparation JCL 126, 343
 - to run programs in VM 511
 - to run SQL programs 493
 - VisualAge Generator parts for execution 521
 - preparing and running programs
 - DB2 129
 - IMS 165
 - MVS batch 159
 - MVS CICS 143
 - MVS/TSO 151
 - preparing for the installation of VisualAge Generator Server for MVS, VSE, and VM 3
 - Preventive Service Planning (PSP) facility 3
 - PRGM00 (Parameter Group List Display panel) 201, 395
 - PRGM00 (Parameter Group Specification panel) 200, 394
 - PRGM02 (Parameter Group Definition panel) 202, 396
 - print destination
 - CICS, specifying in DCT 72
 - default
 - IMS 90
 - MVS/TSO 78, 475
 - VSE CICS
 - default 306
 - specifying in DCT 320
 - print file, utilities 55
 - print services program
 - object of generation 119
 - preloading 94
 - printing
 - American National Standards control character 57, 305
- printing (*continued*)
 - buffer size 60, 308
 - CICS, destination control table (DCT) 58
 - considerations
 - IMS 90
 - MVS CICS 57
 - MVS/TSO 78, 476
 - CREATX call for print destination 58, 306
 - DBCS (double-byte character set)
 - EZEP transaction 306
 - MPP value 309
 - MSGPOPT=CCONTRL option 310
 - printer 320
 - DBCS (double-byte character set), printer 72
 - DCT (destination control table)
 - DBCS considerations 310
 - transient data queue 306
 - transient data queue name 58
 - trigger level 58, 306
 - default
 - message queue file 306
 - print destination 306
 - default, print destination 58
 - destination
 - using CREATX call 306
 - using EZEDESTP special function word 306
 - destination, using CREATX call 58
 - destination control table (DCT)
 - DBCS considerations 310
 - transient data queue 306
 - transient data queue name 58
 - trigger level 58, 306
 - DISPLAY process option 57, 305
 - enqueueing on transient data queue 57, 305
 - EZEP transaction 58, 306, 320
 - EZEPRINT data set 72, 320
 - EZEZ transaction 58
 - file descriptions, MVS CICS 55
 - file descriptions, VSE CICS 303
 - form-feed 58, 306
 - FORMFD=NO parameter 58, 62, 306, 310
 - FZETPRT program 59, 306, 307, 320
 - line character format 57, 305
 - multiple printers 309

- printing (*continued*)
 - MVS CICS
 - considerations 55
 - file descriptions 55
 - new-line order 306
 - page alignment 306
 - parameter
 - FORMFD 59, 62, 307, 310
 - group associated with
 - FZETPRT program 59, 307
 - PRTBUF 59, 307, 308
 - PRTMPP 59, 61, 307, 309
 - PRTTYP 59, 61, 307, 309
 - PCT (program control table),
 - FZETPRT program 62
 - PR01 transient data queue 72, 320
 - print destination, default 58
 - program control table (PCT),
 - FZETPRT program 62, 310
 - PRTBUF parameter 308
 - SEND command 60, 308
 - temporary storage queue 57, 305
 - transient data 321
 - at a terminal device 320
 - queue 306, 320
 - transient data queue 58, 72
 - VSE CICS
 - considerations 323
 - file descriptions 303
 - printing, transient data 73
 - printing, transient data at a terminal device 72, 320
 - printing diagnostic information
 - for IMS 221
 - for MVS CICS 224
 - printing trace output 557
 - problem
 - diagnosis 207, 399, 541
 - reporting sequence 242, 425, 559
 - processing
 - a DISPLAY process option for a printer map 305
 - batch 83
 - SQL programs in VSE batch 330
 - VSE batch 325
 - VSE CICS 303
 - processing mode
 - 24-bit mode 299
 - 31-bit mode 299
 - IMS 90
 - MVS CICS
 - performance 68
 - types 56
 - processing mode (*continued*)
 - performance 316
 - VSE CICS 304
 - processing mode, VSE CICS 304
 - processing program table (PPT)
 - entries 144
 - production libraries, maintaining
 - copies for backup 178, 371, 515
 - profile block
 - map group 229, 413, 551
 - program 229, 413, 551
 - table 229, 413, 551
 - program
 - bind DB2 52, 300, 472
 - COBOL program 338
 - data files, defining 47, 292, 322
 - entries 144
 - interaction errors 211
 - module, preloading 93
 - preloading 94
 - profile block 229, 413, 551
 - return codes 273, 451, 579
 - runtime support for DL/I 81
 - program communication block (PCB)
 - alternate 89
 - ELAPCB macro 89
 - program control table (PCT) 28
 - DTB=YES and DBP value 72, 312
 - entries 145, 349, 351
 - FZETPRT program 62, 310, 312, 320
 - generation output 117
 - interaction errors 403
 - specification block (PSB), defining 347
 - program preparation steps after
 - program generation 485
 - program processing table (PPT) 28
 - entries 349
 - resident parameter 314
 - program specification block (PSB)
 - defining 88
 - generation 88
 - PRTBUF parameter
 - specifying print buffer size 59
 - using with the FZETPRT program 59
 - PRTBUF parameter for FZETPRT
 - program 307
 - PRTMPP parameter
 - specifying maximum print positions 61
 - using with FZETPRT program 61
 - PRTMPP parameter for FZETPRT
 - program 307, 309
 - PRTTYP parameter
 - DBCS printing 61
 - using with the FZETPRT program 59
 - PRTTYP parameter for FZETPRT
 - program 307, 309
 - PSB (program specification block),
 - defining 347
 - pseudoconversational
 - processing mode, MVS CICS 56, 67
 - processing mode, VSE CICS 304, 315
 - programs and residency 67, 315
- ## R
- RCT 148
 - RDO (resource definition online),
 - generation output 117
 - RDO CEDA transaction 145
 - reading, transient data from tape 73, 321
 - reading transient data from tape 321
 - recovery
 - batch considerations 327
 - considerations 327
 - database
 - considerations 311
 - integrity 311
 - integrity, considerations 311
 - SQL considerations 323
 - DL/I, database integrity 324
 - options 311
 - recoverable resource 323
 - SQL, database integrity and recovery 323
 - recovery and restart, VSE batch
 - programs 369
 - recovery considerations
 - DB2
 - IMS 96
 - MVS batch 84
 - MVS CICS 74
 - MVS/TSO 81, 482
 - DL/I
 - IMS 97
 - MVS batch 85
 - MVS CICS 75
 - MVS/TSO 82
 - IMS 90
 - MVS batch 83
 - MVS CICS 63

- recovery considerations *(continued)*
 - MVS/TSO 79, 477
 - recovery options
 - MVS CICS tables 64
 - specifying 64
 - reentrant code 46, 292
 - reentrant load module storage
 - estimates for VisualAge Generator Server for MVS, VSE, and VM 13
 - relational database, recovery 323
 - relative (RRDS) define cluster 47, 293
 - reporting
 - errors 213, 405, 546
 - problems 240, 423, 557
 - request module, DB2 124
 - required files 475
 - RESET service
 - RESET service, MVS batch 84
 - RESET service, MVS CICS 64
 - RESET service routine 327
 - residency
 - considerations
 - IMS 92
 - MVS CICS 65
 - pseudoconversational 315
 - VSE CICS 313
 - general rules, CICS 66
 - general rules, VSE CICS 314
 - VSE CICS 313
 - resident
 - object, VSE CICS 314
 - parameter 66
 - parameter in PPT (processing program table), VSE CICS 314
 - program considerations 356
 - programs 149
 - resource
 - control table 148
 - tables for CICS 143
 - resource control table (RCT) 28
 - Resource Measurement Facility II 31, 95
 - resource tables, modifying, VSE CICS 36
 - response time, improving 94
 - restart, and recovery, VSE batch programs 369
 - RETAIN keyword, GRANT command 129
 - return codes
 - DL/I 447
 - EZERT8 254, 431, 566
 - SQL 266, 445, 573
 - system 251, 431, 565
 - return codes *(continued)*
 - VisualAge Generator Server 256, 435, 567, 579
 - return codes, EZERT8 251
 - RMF 31, 95
 - rollback
 - IMS 91
 - MVS batch 84
 - MVS CICS 64
 - MVS/TSO 80, 478
 - RRDS, data set definition 47, 293
 - running
 - generated programs in VSE CICS 349
 - main program
 - VSE batch 361
 - with DB2/VSE access 363
 - with DB2/VSE and DL/I access 367
 - with DL/I access 365
 - with no database access 363
 - main programs 512
 - main programs under MVS batch 159
 - preparation JCL 126, 343
 - preparation REXX exec 491
 - programs under IMS 170
 - programs under VSE CICS 356
 - VisualAge Generator program on VM 527
 - running under
 - IMS
 - batch as MPPs 173
 - BMP with DB2 175
 - main batch as BMP 174
 - main program under BMP 173
 - MVS batch
 - main batch with DB2 and DL/I 162
 - main batch with DL/I 161
 - main batch with no database 160
 - main batch with no DB2 161
 - MVS CICS 149
 - MVS/TSO 152
 - runtime
 - CLIST 118, 139, 488, 500
 - JCL 118, 137
 - creating 357
 - examples 362
 - modifying 357, 359
 - job stream data set 125
 - messages
 - COBOL 582
 - runtime *(continued)*
 - messages *(continued)*
 - COBOL/370 282, 583
 - IMS 283
 - MVS 285
 - messages, LE 281, 458
 - options, LE 130, 494
 - support for DL/I programs 330
- ## S
- sample CLISTs
 - main program with DB2 access 154
 - main program with DL/I access 156
 - main program with no database access 153
 - sample exec for the Message File Conversion Utility 535
 - sample execs, modifying 497
 - sample JCL
 - BMP with DB2 176
 - IMS BMP program 175
 - MVS Batch with DB2 Access 161
 - MVS Batch with DB2 and DL/I 163
 - MVS batch with DL/I Access 162
 - MVS batch without DB2 160
 - PCT entries 146
 - PPT entries 144
 - RCT entry 148
 - sample runtime JCL
 - /RUNFILE generation option 357
 - input from SYSIPT 329
 - main batch program
 - with DB2/VSE 365
 - with DB2/VSE and DL/I 369
 - with DL/I 367
 - with no DL/I or DB2/VSE 363
 - SEQ files 325
 - template names 358
 - VSE batch 339
 - sample runtime JCL, message file conversion utility 392
 - saving storage space 94
 - scheduling PSBs, ACT entry for DL/I programs 323
 - screen editing errors 207, 399

- security considerations
 - VisualAge Generator Server for MVS, VSE, and VM, IMS 30
 - VisualAge Generator Server for MVS, VSE, and VM, MVS CICS 27
 - VisualAge Generator Server for MVS, VSE, and VM, MVS/TSO 25
 - VisualAge Generator Server for MVS, VSE, and VM, VSE CICS 34
 - VisualAge Generator Server for MVS, VSE, and VM for MVS, VSE, and VM 39
 - VisualAge Generator Server for MVS, VSE, and VM for VM batch 37
 - VisualAge Generator Server for MVS, VSE, and VM for VM CMS 37
- segmented processing mode
 - IMS 90
 - MVS CICS 56
- SEND command, printing 60, 308
- serial (ESDS) define cluster 47, 293
- service module, preloading 93
- services, message format 51
- setting the variable information for ELACCDDEF DATA control file 531
- shared virtual area (SVA), performance considerations 292
- sharing modules 94
- single-segment mode, IMS 90
- SIT (system initialization table) 311
- size, temporary storage for work database 315
- snap dump listing file, IMS 87
- software requirements
 - MVS 7
 - VisualAge Generator Server for MVS, VSE, and VM 7
 - VM 10
 - VSE 9
- specifying
 - print destination, VSE CICS 303
 - recovery options, VSE CICS tables 311
- spool files, MVS CICS 29
- SQL
 - connection to database 322
 - considerations 52, 300, 322, 472
 - database integrity and recovery 323
 - SQL (continued)
 - multiple DB2/VSE program servers 329
 - return codes 266, 445, 573
 - SQL, BIND step for program 128
 - SQL/DS, considerations, for VM CMS, VisualAge Generator Server for MVS, VSE, and VM 37
 - SQL-prepare COBOL programs 521
 - SQL preprocessor, invoking for VisualAge Generator COBOL programs 528
 - starting
 - IMS programs
 - /FORMAT command (transaction) 170
 - directly (main) 170
 - MPPs (transactions) 171
 - SQL programs on VM 481
 - transactions 150
 - VisualAge Generator program on VM 527
 - starting the transaction in VSE CICS 356
 - startup JCL, FCT, VSE CICS 318
 - startup JCL for CICS 148
 - static linkage 128, 346
 - statistics, performance 95
 - status 31
 - codes
 - DL/I 268
 - VSAM 269, 575
 - key values, COBOL 271, 576
 - storage, requests, above the 16MB line 299
 - storage requirements
 - for VisualAge Generator Server for MVS, VSE, and VM COBOL dynamic storage 15
 - subsystem ABEND dumps 228, 412, 550
 - support for DBCS terminals 51, 297, 471
 - SYNCPOINT
 - command, VSE CICS 312
 - ROLLBACK command, VSE CICS 312
 - synonyms for DB2 tables 129, 493
 - synonyms for DB2/VSE tables 347
 - SYSLIN 126
 - SYSLMOD 126
 - SYSOUT system output file 78
 - SYSPRINT system output file 78, 476
- system
 - abend codes 279, 457, 581
 - considerations
 - batch 325
 - VSE CICS 303
 - errors 213, 404
 - output file 78
 - return codes 251, 431, 565
 - system considerations
 - general 45, 291, 467
 - IMS 87
 - MVS batch 83, 475
 - MVS CICS 55
 - printing 57
 - system definition, IMS 31
 - system initialization table (SIT) 311
 - system initialization table (SIT), MVS CICS 28
 - SYSUDUMP system output file 78
- T**
- table
 - modules, preloading 93
 - preloading 94
 - profile block 229, 551
 - program 119, 488
- table profile block 413
- table space
 - expanding 101, 102
 - requirements 101
- tailoring, JCL before generation 358
- tape, reading transient data from 321
- TASKLIB 151
- templates
 - CLIST 135, 141
 - for generating file allocation statements 498
 - JCL 135, 136
 - REXX exec 498
- temporary storage queue 57, 305, 315
- temporary storage queues 29
- terminal
 - printing, VSE CICS 306
- terminal control table (TCT) 29, 72, 320
- terminal printing, MVS CICS 58
- trace facility 230, 414, 552
- trace file 78, 476
- tracing
 - activating 232, 415, 553
 - deactivating 238, 422
- transaction
 - entries 145
 - error 215

- transaction entries (PCT) 351
- TRANSCIVE 29
- transferring control, MVS CICS 64
- transferring control, VSE CICS 312
- transient data
 - printing 321
 - queue
 - TYPE=INTRA entry in DCT 306
 - VSE CICS, defining 320
 - reading from tape 321
 - transient data, defining
 - extrapartition 73, 321
 - transient data, printing 73
 - transient data, printing at a terminal device 72, 320
 - transient data, reading from tape 73
 - transient data queue
 - defining 72
 - printing, MVS CICS 58
 - transient data queue, TYPE=INTRA entry in DCT 58
 - translator step for CICS 127
 - translator step for VSE CICS 345
 - TRMSTAT=TRANSCIVE 29
 - tuning, IMS 95
 - tuning, VSE CICS 317
 - tuning the IMS system 30

U

- UCTRAN 29
- unit of work, logical
 - batch 327, 330
 - IMS 96, 97
 - MVS CICS 64
 - MVS/TSO 79, 477
 - VSE CICS 312
- uppercase translation 53, 302, 474
- user data set 124
- using
 - and allocating VSAM data files in
 - MVS batch 84
 - multiple work databases 103
 - new copy 311
 - remote files, MVS CICS 71
 - remote files, VSE CICS 319
 - remote programs, transactions, or files 354
 - spool files 355
- using generated applications with
 - PL/I programs 41
- using spool files 29
- using the /DATA=24 generation option, VSE CICS 35

- using the /DATA generation option 28
- utilities
 - IMS diagnostic message print 203
 - message file conversion, MVS batch 533
 - new copy 311
- utilities, diagnostic, control options, CICS 181, 375
- utilities, diagnostic, message print utility, CICS 186, 380
- utilities, diagnostic control options, ELAC 189, 383
- utilities, diagnostic message printing utility, ELAU 186, 380
- utilities, message file conversion, batch 195, 389
- utilities, new copy utility, ELAN 183, 377
- utilities, parameter group utility, PRGM 200, 394
- utilities for MVS CICS, VisualAge Generator Server for MVS, VSE, and VM 27
- utility 186, 380
- utility, ELAU 186, 380

V

- virtual lookaside facility (VLF), MVS/ESA 46
- virtual machine size 467
- virtual storage
 - considerations and residency 314
 - constraint relief (VSRC) 299
- virtual storage considerations and residency 66
- virtual storage requirements 13
 - considerations 32
 - IMS 31
 - VisualAge Generator Server for MVS, VSE, and VM 13
- VisualAge Generator Server
 - abend codes 579
 - abend dumps 227, 411, 549
 - availability 151, 511
 - control block 229, 551
 - diagnosing problems 541
 - error 219, 547
 - extended addressing 51
 - return codes 567, 579
 - trace facility for 552
 - VisualAge Generator Server, control options by transaction 190, 384
- VisualAge Generator Server, default control options 193, 387
- VisualAge Generator Server, diagnostic control options 189, 383
- VisualAge Generator Server, new copy 183, 377
- VisualAge Generator Server, utilities, diagnostic control facility, ELAM 181, 375
- VisualAge Generator Server, utilities, diagnostic control options, ELAC 189, 383
- VisualAge Generator Server, utilities, diagnostic message printing utility, 186, 380
- VisualAge Generator Server, utilities, ELAC, diagnostic control options 189, 383
- VisualAge Generator Server, utilities, ELAM, diagnostic control facility 181, 375
- VisualAge Generator Server, utilities, ELAN, new copy utility 183, 377
- VisualAge Generator Server, utilities, ELAU, diagnostic message printing 186, 380
- VisualAge Generator Server, utilities, new copy utility, ELAN 183, 377
- VisualAge Generator Server, utilities, parameter group utility, PRGM 200, 394
- VisualAge Generator Server, utilities, PRGM, parameter group utility 200, 394
- VisualAge Generator Server for MVS, VSE, and VM
 - application load module storage 14
 - applying maintenance 3
 - client/server processing
 - considerations for MVS CICS 27
 - client/server processing for VSE CICS 35
 - COBOL dynamic storage 15
 - COBOL external storage for non-CICS environments 15
 - customizing JCL procedures 41
 - DB2 considerations for IMS 30
 - DB2 considerations for MVS CICS 26
 - DB2 considerations for MVS/TSO 25

- VisualAge Generator Server for MVS, VSE, and VM *(continued)*
 - DB2/VSE considerations for VSE batch 33
 - DB2/VSE considerations for VSE CICS 34
 - DB2 work database considerations for IMS 32
 - disk storage requirements 18
 - DL/1 considerations for MVS/TSO 24
 - DL/1 considerations for VSE batch 32
 - DL/1 considerations for VSE CICS 34
 - DL/I considerations for MVS CICS in VisualAge Generator Server for MVS, VSE, and VM 26
 - dynamic storage 16
 - hardware requirements 5
 - IMS/ESA exploitation 29
 - installation considerations for IMS 29
 - installation considerations for MVS batch 24
 - installation considerations for MVS CICS 26
 - installation considerations for MVS/TSO 24
 - installation considerations for VSE CICS 33
 - load module storage, reentrant 13
 - load modules storage 13
 - making generated programs available for MVS/TSO 25
 - making modules available for MVS/TSO 25
 - modifying MVS CICS resource tables 28
 - modifying VSE CICS resource tables 36
 - module storage estimates, statically linked 14
 - performance considerations for MVS, VSE, and VM 40
 - preparing for the installation of 3
 - security considerations for IMS 30
 - security considerations for MVS, VSE, and VM 39
 - security considerations for MVS CICS 27
- VisualAge Generator Server for MVS, VSE, and VM *(continued)*
 - security considerations for MVS/TSO 25
 - security considerations for VM CMS 37
 - security considerations for VSE CICS 34
 - software requirements 7
 - specifying the WSCLEAR option for COBOL 41
 - SQL/DS considerations for VM CMS 37
 - storage requirements 13
 - temporary storage queues 29
 - use of MVS CICS storage facilities 17
 - using generated applications with PL/1 programs 41
 - utilities for MVS CICS 27
 - utilities for VSE CICS 35
 - virtual storage requirements 13
 - VSE batch installation considerations 32
 - work database space for segmented applications 18
- VisualAge Generator Server for MVS, VSE, and VM,
 - installation considerations, VM batch 37
 - installation considerations, VM CMS 37
 - security considerations for VM batch 37
 - security considerations for VM CMS 37
- VM
 - VisualAge Generator Server for MVS, VSE, and VM installation 3
 - VM, VisualAge Generator Server, software requirements 10
 - VM batch, installation considerations, VisualAge Generator Server for MVS, VSE, and VM 37
 - VM batch, security considerations, VisualAge Generator Server for MVS, VSE, and VM 37
 - VM CMS, installation considerations 37
 - VM CMS, installation considerations, VisualAge Generator Server for MVS, VSE, and VM 37
- VM CMS, security considerations, VisualAge Generator Server for MVS, VSE, and VM 37
- VM CMS, SQL/DS considerations 37
- VS COBOL II messages 582
- VSAM
 - data set definition 47, 292, 293
 - defining an alternate index 47, 293
 - defining data files 468
 - DEFTTEST AMSERV 469
 - file loading 50, 296
 - indexed (KSDS) data set 47, 293
 - loading data into a file 470
 - managing data files 468
 - program data files 478
 - relative (RRDS) data set 47, 293
 - serial (ESDS) data set 47, 293
 - status codes 269, 448, 575
- VSE
 - VisualAge Generator Server for MVS, VSE, and VM installation 3
 - VSE, VisualAge Generator Server, software requirements 9
 - VSE batch
 - ACT entry for MPS batch with DL/I databases 331
 - activating trace sessions 419
 - database integrity and recovery considerations, SQL 330
 - DB2/VSE, EZECONCT service routine 329
 - DL/I
 - ACT entry MPS batch with DL/I databases 331
 - considerations 330
 - examples of runtime JCL 362
 - EZECONCT service routine 329
 - logical unit of work (LUW)
 - description 327
 - recoverable resource 330
 - LUW (logical unit of work)
 - description 327
 - recoverable resource 330
 - preparing, generated programs 357
 - program runtime support for DL/I 330
 - recoverable resource, logical unit of work (LUW) 330
 - recovery and restart, batch programs 369
 - recovery considerations 327

VSE batch (*continued*)
 RESET service routine 327
 rollback 328
 running
 main program with DB2/VSE
 access 363
 main program with DB2/VSE
 and DL/I 367
 main program with DL/I
 access 365
 main program with no
 database access 363
 running generated programs
 357
 VSE/POWER, print services
 program 339

VSE CICS
 abend
 ASPE abend code 312
 recovery considerations 323
 activating trace sessions 416
 ASPE abend 312
 CALL statement 312
 CICSPARS (performance analysis
 reporting system) 317
 commit point 312
 CONNECT statement 322
 controlling diagnostic
 information 356
 conversational processing mode
 304
 data queue
 extrapartition 320
 intrapartition 320
 database
 connection for SQL programs
 322
 integrity considerations 311,
 323
 recovery considerations 311,
 323
 relational database recovery
 considerations 323
 database temporary storage size
 315
 DBCS printer 320
 DCT (destination control table)
 entries 354
 printing, DBCS 310
 sample entry for EZEP
 transaction 320
 terminal entry for printing
 320
 transient data queue 320

VSE CICS (*continued*)
 defining
 data file 317
 intrapartition transient data
 files 320
 destination control table (DCT)
 entries 354
 printing, DBCS 310
 sample entry for EZEP
 transaction 320
 terminal entry for printing
 320
 transient data queue 320

DL/I
 ACT (application control
 table) 323
 application control table
 (ACT) 323
 DBP value 312
 dynamic backout support
 312
 programs that schedule PSBs
 323
 recovery and database
 integrity 324
 DXFR statement 312
 dynamic backout 312
 EZECONCT service routine 322
 EZEP transaction 306, 320
 EZEROLLB service routine 312
 EZESEGM special function word
 304
 EZEZ transaction 306, 320
 FCT (file control table)
 data set name, user 318
 entries 318, 354
 entry, relationship to DLBL in
 JCL 318
 file control table (FCT)
 data set name, user 318
 entries 318, 354
 entry, relationship to DLBL in
 JCL 318
 file descriptions 303
 JCL
 DLBL name and FCT entry
 318
 startup 318
 LINK statement 312
 logical unit of work (LUW)
 committing changes,
 recoverable resources 312
 description 324
 ending 312

VSE CICS (*continued*)
 LUW (logical unit of work)
 committing changes,
 recoverable resources 312
 description 324
 ending 312
 making new modules available
 355
 mode
 conversational processing
 304
 nonsegmented processing
 304
 pseudoconversational,
 residency consideration
 315
 pseudoconversational,
 segmented processing 304
 segmented processing 304
 modifying startup JCL 355
 monitoring and tuning 317
 new copy 311
 nonsegmented processing mode
 304
 partition, database access 322
 PCT (program control table)
 DTB=YES and DBP value
 312, 320
 entries 351
 entry 310, 320
 printing, DBCS printing 306,
 310
 performance
 monitoring 317
 processing mode 316
 resident object 313
 performance analysis reporting
 system (CICSPARS) 317
 PPT (processing program table)
 entries 349
 entry 312, 320
 FN=BACKOUT 312
 resident parameter 314
 TYPE=GROUP 312
 preparing programs 349
 printing
 American National Standards
 control character 305
 buffer size 308
 DBCS (double-byte character
 set) 306, 309, 310
 DCT (destination control
 table) 306, 310, 320
 destination, using CREATX
 call 306

- VSE CICS (*continued*)
 - printing (*continued*)
 - destination, using EZEDESTP special function word 305
 - destination control table (DCT) 306, 310, 320
 - DISPLAY process option 305
 - enqueueing on transient data queue 305
 - EZEP transaction 306, 320
 - EZEPRINT data set 320
 - EZEZ transaction 306, 320, 321
 - file description 303
 - form-feed 306
 - FORMFD=NO parameter 306
 - FZETPRT program 306, 307, 320
 - line character format 305
 - multiple printers 309
 - new-line order 306
 - page alignment 306
 - parameter, PRTTYP 309
 - PCT (program control table) 320
 - program control table (PCT) 310, 320
 - PRTBUF parameter 307, 308
 - PRTMPP parameter 307
 - PRTTYP parameter 307
 - SEND command 308
 - TCT (terminal control table) 320
 - temporary storage queue 305
 - terminal control table (TCT) 320
 - transient data 321
 - transient data queue 305, 320
 - processing mode 316
 - performance 316
 - types 304
 - processing program table (PPT)
 - entries 349
 - entry 312, 320
 - FN=BACKOUT 312
 - resident parameter 314
 - TYPE=GROUP 312
 - processing type (see mode) 304
 - program and transaction
 - definitions file 341
 - program control table (PCT)
 - DTB=YES and DBP value 320
 - entries 351
 - VSE CICS (*continued*)
 - program control table (PCT) (*continued*)
 - entry 320
 - printing, DBCS printing 306
 - pseudoconversational
 - processing mode 304, 315
 - programs and residency 315
 - recoverable resource, recovery considerations 323
 - relational database, recovery considerations 323
 - RESET service routine 312
 - residency
 - considerations 313, 315
 - general rules 314
 - resident
 - object 314
 - program considerations 356
 - rollback 312
 - running
 - a generated program 349
 - programs 356
 - SIT (system initialization table) 311
 - SQL
 - database connection 322
 - recovery considerations 323
 - starting the transaction 356
 - SYNCPOINT
 - command 312
 - ROLLBACK command 312
 - system considerations 303
 - system initialization table (SIT) 311
 - TCT (terminal control table) 320
 - temporary storage
 - queue 305
 - size 315
 - terminal control table (TCT) 320
 - terminal printing 306
 - transaction
 - EZEP 306, 320
 - EZEZ 306, 320
 - PR01 transient data queue 320
 - transferring control
 - CALL statement 312
 - DXFR statement 312
 - LINK statement 312
 - XCTL statement 312
 - translator step 345
 - tuning 317
 - type of processing 304
 - VSE CICS (*continued*)
 - using
 - remote programs, transactions, or files 354
 - spool files 355
 - XCTL statement 312
 - VSE CICS, installation considerations 33
 - VSE CICS, printing, transient data 73
 - VSE CICS utilities, VisualAge Generator Server for MVS, VSE, and VM 35
 - VSE/POWER
 - formatting data to the printer 339
 - print services program, VSE batch 339
- ## W
- warnings
 - empty KSDS data set, VSAM restriction 50, 296
 - PRTMPP parameter, line skip malfunction 61, 309
 - work, database, temporary storage size 315
 - work database
 - clearing records 98
 - deleting old records 98
 - expanding 101
 - maintaining 98
 - multiple 103
 - temporary storage size 68
 - work database, ELAPCB macro 89
 - work database, temporary storage size 315
 - work database space for segmented applications, VisualAge Generator Server for MVS, VSE, and VM 18
 - WORK parameter in ELAPCB 89
 - WSCLEAR option for COBOL 41
 - WSCLEAR option for COBOL, specifying 41
- ## X
- XCTL, transferring control in CICS 64
 - XCTL, transferring control in VSE CICS 312

Readers' Comments — We'd Like to Hear from You

VisualAge Generator Server Guide for MVS, VSE, and VM
Version 1.2

Publication No. SH23-0256-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

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

Name

Address

Company or Organization

Phone No.



Cut or Fold
Along Line

Fold and Tape

Please do not staple

Fold and Tape



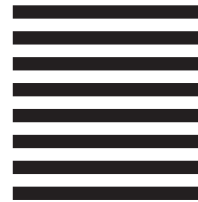
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Information Development
Department G71A / Bldg 062
P.O. Box 12195
Research Triangle Park, NC
27709-2195



Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold
Along Line



Program Number: 5648-B02



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SH23-0256-00

