



Migrating Cross System Product Applications to VisualAge Generator

Version 3.1



Migrating Cross System Product Applications to VisualAge Generator

Version 3.1

Note

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

Second Edition (November 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 1980, 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	vii
Trademarks	ix
Terminology Used in This Document	ix
About This Document.	xiii
Who Should Use This Document	xiii
Documentation provided with VisualAge Generator	xiii
Related information	xiv

Part 1. Migration Process

Overview	1
Chapter 1. Migration Process	3
ENVY Characteristics	3
Comparison of MSLs and ENVY	5
Member Types	5
Storing Members	6
Storing Control Information	6
MSL Concatenation	8
Functional Organization	8
Migration Suggestions	9

Part 2. Migrating from Cross System Product 3.2.1 or Earlier 11

Chapter 2. CSP/AD 3.2.1 or Earlier to VisualAge Generator Developer	13
MSLs	13
Considerations for Migrating an MSL from VSE/ESA to the Workstation	13
Message File Conversion	17
Test Facility Limitations	17
Testing TSO or CICS Programs	17
Definition Limitations	18
Special Function Words	18
Generation Options	19
Resource Associations	19
Migrating SETGEN Information	19
Tables	20
Linkage Types	20

Chapter 3. CSP/AE 3.2.1 or Earlier to VisualAge Generator Server for MVS, VSE, and VM or VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX	21
---	-----------

Part 3. Migrating from Cross System Product 3.2.2 or 3.3 . . . 23

Chapter 4. CSP/AD 3.2.2 or CSP/AD 3.3 to VisualAge Generator Developer	25
MSLs	25
Considerations for Migrating an MSL from VSE/ESA to the Workstation	25
Message File Conversion	28
Test Facility Limitations	28
Testing TSO or CICS Programs	28
Definition Limitations	29
Special Function Words	30
Generation Options	30
Resource Associations	30
Migrating SETGEN Information	31
Tables	31
Linkage Types	31

Chapter 5. CSP/AD 3.3 Programmable Workstation Feature to VisualAge Generator Developer	33
CSP/AD 3.3 Programmable Workstation Feature MSLs	33
Definition Limitations	33
Batch Commands	34
NLS Parameters	34

Chapter 6. CSP/AE 3.2.2 or 3.3 to VisualAge Generator Server for MVS, VSE, and VM or VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX	35
---	-----------

Chapter 7. CSP/370RS COBOL Generation to VisualAge Generator Developer	37
Generation Options	37

Generation Options in the MSL	37
Generation Options File	37
Resource Associations	38
Migrating SETGEN Information	39
Tables	39
Linkage Types	40
Command Interface	40
Reserved Word File	40
Templates	40
Generation Output Libraries on the Host	42
Interfacing to PL/I	43
SQL Plan Definition	43
Chapter 8. CSP/370RS 1.1 to VisualAge	
Generator Server for MVS, VSE, and VM	45
Installation Considerations	45
Procedures	45
Upward Compatibility	46
Generating Applications, Tables, and	
Map Groups Again	46
Generated Tables	47
Applications Accessing the Same Table	47
Error Routines	47
Defining PSBs	47
SQL Plan Definition	48
User Messages	48

Part 4. Migrating from Cross System Product 4.1 49

Chapter 9. CSP/370AD 4.1 to VisualAge	
Generator Developer	51
MSLs	51
Message File Conversion	51
Test Facility Limitations	52
Testing TSO or CICS Programs	52
Definition Limitations	52
Generation Options	53
Generation Options in the MSL	53
Generation Options File	53
Resource Associations	54
Migrating SETGEN Information	54
Tables	55
Linkage Tables	55
Command Interface	55
Reserved Word File	55
Templates	56
Generation Output Libraries on the Host	58

Chapter 10. CSP/2AD 1.2 to VisualAge	
Generator Developer	59
CSP/2AD 1.2 MSLs	59
Defintion Limitations	59
Batch Commands	59
NLS Parameters	59

Chapter 11. CSP/370RS 2.1 to VisualAge	
Generator Server for MVS, VSE, and VM	61
Installation Considerations	61
Procedures	62
Upward Compatibility	62
Generating Applications, Tables, and	
Map Groups Again	62

Chapter 12. CSP/2RS 1.1 to VisualAge	
Generator Server for OS/2, AIX, Windows	
NT, and HP-UX	65
CSP/2RS File Migration Considerations	65
Upward Compatibility	67
OS/2 CICS User Data Files	68

Chapter 13. Running CSP/370AD 4.1 or	
VisualAge Generator 2.2 Applications	
with VisualAge Generator Server for	
OS/2, AIX, Windows NT, and HP-UX 3.0 or	
Later.	69

Part 5. Appendixes 71

Appendix A. Language Differences from	
the Cross System Product Set	73
Language Differences from CSP/370AD 4.1	
to VisualAge Generator	73
New and Enhanced Function Words	73
New String-Handling Function Words	73
New Math Function Words	74
Client/Server Processing Enhancements	76
Cursor with Hold Option	76
Maximums Increased	76
Table Specification Changes	76
Trace Facility Enhancement	77
Host Product Upgrades	77
CALL Statement Enhancements	77
Language Differences from CSP/AD 3.3 to	
VisualAge Generator	77
Transactional Processing	77
System Resource Association	78
Client/Server Processing Support	80

Processing Statement Extensions	80
Date Edit Mask for Map and Data Item Definition	81
Set Extended Attributes	81
SQL Dynamic Table Access	81
EZECONCT Option	81
Passing of PCBs in EZEDLPCB	81
User Message Files as Tables	81
Language Differences from CSP/AD 3.2.2 to VisualAge Generator	82
Appendix B. Cross System Product	
Interpretive to COBOL	83
Part Names	83
Language Element Compatibility Considerations	84
Mapping Support	84
CONVERSE Process Option	85
EZE Special Function Words	86
Transferring Among Programs	87
Programming Statements	87
File Support	88
Relational Database (SQL) Support	90
DL/I Database Support	91
Data Values	91
User Message Files	93
Language-Dependent Variables	93
CSP/AD Language Elements Associated with the Application Load File	94
Application Storage Requirements	94
Coexistence with CSP/AE Applications	95
Calls or Transfers from Non-CSP Programs	96
Transferring Control in the CICS Environment	96
MVS CICS and VSE CICS Resource Table Requirements for Programs	97
MVS CICS and VSE CICS Table Entries for VisualAge Generator Server for MVS, VSE, and VM.	97
National Language Codes	98
CSP/AE Invocation Parameters	98
MVS CICS Invocation Parameters	98
MVS/TSO and MVS Batch Invocation Parameters	99
VSE CICS and VSE Batch Invocation Parameters	100
VM Invocation Parameters	102
National Language Applications	104

Non-VisualAge Generator Interface to DCBRINIT on MVS CICS and VSE CICS CONVERSEs	104
--	-----

Appendix C. Cross System Product	
Interpretive to C++	111
Part Names	111
Language Element Compatibility Considerations	111
Mapping Support	111
CONVERSE Process Option	112
EZE Special Function Words	112
Transferring Among Programs	113
File Support	113
Relational Database (SQL) Support	114
DL/I Database Support	115
Data Values	115
User Message Files	115
Transferring Control in the CICS Environment	116
National Language Codes	116
CSP/AE Invocation Parameters	117

Appendix D. Host Environments to	
VisualAge Generator Server for OS/2,	
AIX, Windows NT, and HP-UX	
Conversion of Host Data Files	119
SQL Considerations	120

Appendix E. Migrating to VisualAge	
Generator with TeamConnection	
Source Code	121
Linkage Table Part	121
Resource Association Part	122

Appendix F. Using the Message File	
Conversion Utility	
123	
Using CSP/AE 3.3 Message File Conversion Utility	123
Return Codes	124
Message Table Conversion Parameters	125
Sample MVS JCL for CSP 3.3 Message File Conversion Utility	126
Sample VSE JCL for CSP 3.3 Message File Conversion Utility	127
Using VisualAge Generator Message File Conversion Utility	127
Converting a Message File to an External Source Format Table.	128

Appendix G. Host Generation Option
Changes 129

Index 131

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:

AIX	IMS/ESA
AD/Cycle	Language Environment/370
CICS	MVS
CICS/ESA	Operating System/2
CICS OS/2	OS/2
CICS/MVS	OS/400
CICS/VSE	SAA
CICS/6000	SAA AD/Cycle
COBOL/2	SQL/DS
COBOL/370	SQL/400
COBOL/400	TeamConnection
DB2/2	Virtual Machine/Enterprise Systems Architecture
DB2	VisualAge Generator
DB2/400	VisualGen
DRDA	VM/ESA
IBM	VS COBOL II
ILE	VSE/ESA
IMS	VTAM

The following terms are trademarks of other companies:

C++	American Telephone & Telegraph Company
HP-UX	Hewlett-Packard Company
Micro Focus COBOL	Micro Focus Limited

ENVY is a registered trademark of Object Technology International Inc.

Microsoft, Windows, Windows NT, Visual Basic, 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.
- Unless otherwise noted, references in this publication to Cross System Product apply as follows:

CSP/AD 3.2.2

IBM Cross System Product/Application Development Version 3
Release 2 Modification 2

CSP/AD 3.3

IBM Cross System Product/Application Development Version 3
Release 3

CSP/370AD 4.1

IBM SAA Cross System Product/370 Application Development
Version 4 Release 1

CSP/AD 1.2

IBM SAA Cross System Product/2 Application Development
Version 1 Release 2

CSP/AE 3.2.2

IBM Cross System Product/Application Execution Version 3 Release
2 Modification 2

CSP/AE 3.3

IBM Cross System Product/Application Execution Version 3 Release
3

CSP/370RS 1.1

IBM SAA Cross System Product/370 Runtime Services Version 1
Release 1

CSP/370RS 2.1

IBM SAA Cross System Product/370 Runtime Services Version 2
Release 1

CSP/2RS 1.1

IBM SAA Cross System Product/2 Runtime Services Version 1
Release 1

About This Document

This document provides considerations for migrating Cross System Product applications to VisualAge Generator.

Who Should Use This Document

This document is for Cross System Product developers or administrators who are migrating their applications to VisualAge Generator Version 3.1 or later.

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

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

- *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:

- *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* (SH23-0244-01)
- *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 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 Smalltalk:

- *VisualAge Smalltalk Getting Started* (SC34-4535)
- *VisualAge Smalltalk User's Guide* (SC34-4518)
- *VisualAge Smalltalk User's Reference*
- *IBM Smalltalk Programmer's Reference* (SC34-4493)
- *IBM Smalltalk User's Guide* (SC34-4536)
- *VisualAge Smalltalk Image Component Developer's Guide and Reference*
- *VisualAge Smalltalk AS/400 Connections User's Guide*
- *VisualAge Smalltalk Communications/Transactions Guide and Reference*
- *VisualAge Smalltalk Distributed Guide and Reference*
- *VisualAge Smalltalk Web Connection User's Guide*

- *VisualAge Smalltalk Features Class Guide*
- *VisualAge Smalltalk CICS & IMS Connection User's Guide and Reference*
- *VisualAge Smalltalk Connection for Lotus Notes*
- *VisualAge Smalltalk Installation Guide*
- *VisualAge Smalltalk Multimedia Guide and Reference*
- *VisualAge Smalltalk Programmer's Guide to Building Parts for Fun and Profit (SC34-4496)*
- *VisualAge Smalltalk Reports Guide and Reference*
- *VisualAge Smalltalk Server Enablement for MVS/ESA User's Guide (SC34-4615)*
- *VisualAge Smalltalk Server for MVS/ESA User's Guide and Reference (SC34-4618)*
- *VisualAge Smalltalk Tivoli Connection Guide and Reference*
- *VisualAge Smalltalk Visualization Tools User's Guide*
- *VisualAge Smalltalk Bidirectional Languages Guide*
- *VisualAge Smalltalk Migration Guide*
- *VisualAge Smalltalk ObjectExtender User's Guide and Reference*
- *VisualAge Smalltalk Server Guide (SC34-4740)*
- *VisualAge Smalltalk UML Designer 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. Migration Process Overview

Chapter 1. Migration Process	3
ENVY Characteristics	3
Comparison of MSLs and ENVY	5
Member Types	5
Storing Members.	6
Storing Control Information	6
MSL Concatenation	8
Functional Organization	8
Migration Suggestions	9

2 Migrating to VisualAge Generator

Chapter 1. Migration Process

In both Cross System Product and releases of VisualAge Generator prior to V3.0, code was written in small pieces called members. Members were stored in Member Specification Libraries (MSLs). In VisualAge Generator V3.0 or later, the code must be stored in ENVY, a library management system. If you plan to use the code from Cross System Product and previous releases of VisualAge Generator, you must migrate this code from the MSLs to ENVY. This chapter explains the following:

- “ENVY Characteristics”
- “Comparison of MSLs and ENVY” on page 5

ENVY Characteristics

Using the ENVY library manager to store information is a major change for VisualAge Generator V3.0 and later. There are new terms that are important for the ENVY environment. Because ENVY was originally developed for use with Smalltalk, some explanations of the terms include Smalltalk information to help relate the terminology to Smalltalk. The following terms are new for VisualAge Generator V3.0 and later:

New Term

Relationship to MSL Concepts

VAGen part class

Each 4GL member type (all member types except GUIs) becomes a **VAGen part class**. A VAGen part class is an **extension of a class** in Smalltalk. The VAGen part classes appear in the **Parts** pane of the **VisualAge Organizer** window. The VAGen part classes created for the member types are prefixed by *VAGen* (for example, *VAGenRecords*).

There are five additional VAGen part classes that are used to contain control information that was stored outside the MSL in previous releases of VisualAge Generator. These VAGen part classes are for linkage table, resource association, generation options, bind, and linkage editor information.

VAGen part

Each 4GL member is now stored as a **VAGen part**. A VAGen part is associated with a Smalltalk **method** in an extension of its

VAGen part class. The VAGen parts appear in the **VAGen Parts** pane of the **VisualAge Organizer** window and in the **VAGen Parts Browser** window.

View	Each GUI is now stored as a view , which is a visual part . A view is a class in Smalltalk. The views appear in the Parts pane of the VisualAge Organizer window. Views do not appear on the VAGen Parts Browser in ENVY.
Program	The application member type has been changed to program to distinguish it from an ENVY application.
ENVY Application	An ENVY application is a group of classes and methods that are closely related in function. An ENVY application can include VAGen part classes and VAGen parts. An ENVY application is also called an application .
Configuration Map	A configuration map is a group of application editions that should be loaded together into a developer's image.

The following ENVY concepts are new for VisualAge Generator V3.0 and later:

Concept	Description
Functional Organization	ENVY enables you to group parts into applications. These applications can (and should) be organized along functional lines.
Ownership	<p>Each configuration map and each application has an assigned manager who is responsible for the integrity of the code that is placed in the configuration map or application.</p> <p>Each part (class) or VAGen part class has an assigned owner who is responsible for the integrity of the code that is placed in the class. For 4GL parts, this means that the owner of the class <i>VAGenRecords</i> within application XYZ is responsible for the integrity of all record definitions stored as part of application XYZ. Because each GUI becomes a separate view (visual part), each GUI within application XYZ</p>

can have a different owner. 4GL parts used within the GUI become VAGen parts.

Note: In releases of VisualAge Generator prior to V3.0, the closest concept to ownership was write-protecting the staging, test, or production MSLs and only giving a team leader the authority to advance members into these MSLs.

Edition

Each change that is made to a 4GL VAGen part results in a new edition of the VAGen part being stored in the ENVY library manager. Editions of parts, applications, and configuration maps are also stored in the ENVY library manager.

Version

Editions can be frozen to prevent further changes to that level of code. The frozen edition is called a **version**. After a part, application, or configuration map is versioned, the only way to make changes is to open a new edition.

Image

An image is the developer's current view of the ENVY library manager. It contains the version or edition of the configuration maps, applications, and parts that the developer wants to work on. Only one copy of a VAGen part can be loaded into the image at one time.

Comparison of MSLs and ENVY

This section describes how concepts you are familiar with for MSLs relate to concepts in the ENVY library manager.

Member Types

In Cross System Product and releases of VisualAge Generator prior to V3.0, there was one member type for each type of code that could be written.

With VisualAge Generator V3.0 or later, each 4GL member type is a VAGen part class that is prefixed with *VAGen* (for example, *VAGenRecords*). For GUIs, there is no corresponding VAGen part class because each GUI becomes a separate class.

Table 1 shows the correspondence between member types and VAGen part classes.

Table 1. Member Types and VAGen Part Classes

Member Type	VAGen Part Class
Application	VAGenPrograms
GUI (for VisualAge Generator only)	Not Applicable (GUIs become views)
Record	VAGenRecords
Table	VAGenTables
Data Item	VAGenDataItems
Map Group	VAGenMapGroups
Map	VAGenMaps
Process	VAGenProcesses
Statement Group	VAGenStatementGroups
PSB (Program Specification Block)	VAGenPSBs

Storing Members

In releases of VisualAge Generator prior to V3.0, an MSL was an OS/2 directory and each member was a file within the directory. In Cross System Product, an MSL was a VSAM file and each member was stored as records within the file.

With VisualAge Generator V3.0 or later, all information is stored in the ENVY library manager. Each 4GL member is a VAGen part and is associated with a Smalltalk method. Each GUI is a view (visual part) and is a Smalltalk class.

Storing Control Information

For Cross System Product and releases of VisualAge Generator prior to V3.0, most control information related to test and generation was stored outside the MSL. This control information included:

- Generation options that indicate how an application is to be generated. For example, generation options control whether working storage records are to be initialized, what high level qualifier on the host is to be used for preparation, and what linkage table is to be used. For Cross System Product, only COBOL generation used generation options and these were stored in separate files outside the MSL. For releases of VisualAge Generator prior to V3.0, COBOL generation options were stored in separate files.
- Linkage table that indicates how a CALL, DXFR, or XFER is to be implemented. For the CICS target environment, the linkage table is also

used to indicate whether a VSAM or transient data queue is to be accessed locally or remotely. For example, the linkage table might specify that a CALL to application XYZ in the MVS CICS target environment is to be implemented as a remote call passing data in the CICS COMMAREA. For Cross System Product prior to V4.1, there was no linkage table. For Cross System Product V4.1 and releases of VisualAge Generator prior to V3.0, the linkage table was in a separate file.

- Resource association file that indicates for a specific file how it is to be implemented in a specific target environment. For example, a serial file in the MVS CICS target environment might be implemented as a VSAM file, as a transient data queue, as a temporary storage queue, or as a CICS spool file. For Cross System Product, resource association information was stored in the MSL. For releases of VisualAge Generator prior to V3.0, resource association information was stored in a resource association file outside the MSL.
- Bind control commands that provide information needed for binding the DB2 application plan on an MVS system. For Cross System Product/AE, bind control information was not used. Cross System Product COBOL generation used bind control commands and these were stored in separate files outside the MSL. For releases of VisualAge Generator prior to V3.0, bind control commands were stored in separate files.
- Linkage editor control statements that provide linkage editor information for MVS, VSE, and VM systems. For Cross System Product/AE, linkage editor control statements were not used. Cross System Product COBOL generation used linkage editor control statements and these were stored in separate files outside the MSL. For releases of VisualAge Generator prior to V3.0, linkage editor control statements were stored in separate files.

With VisualAge Generator V3.0 or later, generation options files, linkage tables, resource association files, bind control commands, and linkage editor control statement files are VAGen part classes in ENVY. Thus all the data required for test and generation is contained in a single library management system.

Table 2 shows the correspondence between the types of control information and VAGen part classes.

Table 2. Control Information and VAGen Part Classes

Control Information	VAGen Part Class
Generation Options	VAGenOptions
Linkage Table	VAGenLinkages
Resource Associations	VAGenResources
Bind Control Commands	VAGenBindControls

Table 2. Control Information and VAGen Part Classes (continued)

Control Information	VAGen Part Class
Linkage Editor Control Statements	VAGenLinkEdits

MSL Concatenation

In Cross System Product and releases of VisualAge Generator prior to V3.0, MSLs could be concatenated. Specifying the MSL concatenation sequence was the way in which you specified where to look for the members needed by your applications. When MSLs were concatenated, for test and generation, only the first found member with a given name was used. For viewing, members from MSLs other than where the first found member was located could be referenced. If changes were made to a member in the MSL concatenation sequence, the changed member was stored in the read/write MSL (first MSL in the concatenation sequence).

With VisualAge Generator V3.0 or later, there is no concept similar to MSL concatenation. All editions and versions are available in the ENVY library manager. However, only one edition or version of a part can be loaded into your image at a time. Browsers are available to compare editions and versions within the ENVY library manager before loading them into your image to determine which one is the required level of code.

Configuration maps can also be used to group code for a particular level. For example, you might have a configuration map for production that indicates the version of each application that is in production. Specifying prerequisite maps for a configuration map is similar to specifying the MSL concatenation sequence. The prerequisite maps provide a way for you to ensure that all the parts needed to run a particular group of programs are loaded into your image.

Functional Organization

In Cross System Product and releases of VisualAge Generator prior to V3.0, members were grouped together into MSLs. Generally, an MSL contained all the members for a particular subsystem. Because the MSL was the only method for grouping members by function, the functions tended to be quite large. In Cross System Product the number of MSLs in a concatenation sequence was limited to 6. This also contributed to having a large number of members in each MSL.

With VisualAge Generator V3.0 or later, the configuration map and the ENVY application provide a two-level capability for grouping parts. The configuration map is the higher level of organization and more closely resembles an MSL in terms of the number of parts. ENVY applications enable

you to organize your parts into smaller groups than was reasonable to do with MSLs. This provides more capabilities in terms of controlling access to the parts, finding a part, and limiting the number of parts displayed in the **VAGen Parts Browser**.

Migration Suggestions

This section provides some suggestions for a process to migrate Cross System Product applications to VisualAge Generator.

- Determine what development environment you are going to use (for example, OS/2 or Windows NT).
- Determine if you have the equipment to run VisualAge Generator.
- Determine how you should set up your test environment. For example, will the data stay on the host or will the data be on the workstation for testing. This will affect what hardware and software you need. You need to decide how you will handle your non-Cross System Product programs for the test environment.
- Determine how you want to handle database access such as SQL, DL/I, and VSAM. Decide how you want to handle VSAM access and what to do with SQL and its need for qualifiers.
- Determine if your Cross System Product applications use DL/I. If so, you need to purchase the appropriate Microfocus IMS products so you can test the applications on the workstation.
- Determine if your Cross System Product applications are using dynamic or static SQL. VisualAge Generator does not have an option to use dynamic SQL as a default. If you used the default dynamic SQL instead of static SQL in Cross System Product, you will need to determine how you want to set the permissions on the static plans in DB2.
- Determine what CICS PPT and PCT entries will be required for each program and get the system programming staff to update CICS with these entries.
- Install and test VisualAge Generator Server for MVS, VSE, and VM and VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX completely, including researching the PSP buckets for needed PTFs. For CICS, ensure that the runtime services utility, ELAM, is installed and working correctly. You will need this utility for problem determination.
- Convert your MSLs from Cross System Product to the VisualAge Generator library management system. Refer to the *VisualAge Generator Guide to Migrating MSLs to ENVY* for information on how to migrate your MSLs to the ENVY library management system.
- Keep the same characters to define constant and variable fields when you import Cross System Program maps into VisualAge Generator. Typically, these characters are the "number/pound" sign (#) character and the "not"

character (uppercase 6 on a 3270 keyboard). The "not" character is not found on personal computer (PC) keyboards. However, you do not need to be concerned about this, because the VisualAge Generator Map Editor does not use constant or field delimiters.

- Validate the VisualAge Generator programs. This provides a list of errors where Cross System Product rules were violated but were not caught until you migrated to VisualAge Generator. Use this list to correct the errors.
- Copy all preparation and runtime templates for your environment (MVS, VSE or VM) to a separate subdirectory and make the necessary modifications as needed to this copy. You might want to start by generating and preparing a simple one-map program. Then use the errors reported on the initial preparation to determine how to change the templates to match your environment.
- If you are migrating from interpretive Cross System Product, generate all the VisualAge Generator programs and test them in the runtime environment.

Part 2. Migrating from Cross System Product 3.2.1 or Earlier

Chapter 2. CSP/AD 3.2.1 or Earlier to VisualAge Generator Developer 13
MSLs. 13
Considerations for Migrating an MSL from VSE/ESA to the Workstation 13
Message File Conversion 17
Test Facility Limitations 17
 Testing TSO or CICS Programs 17
Definition Limitations 18
Special Function Words 18
Generation Options 19
Resource Associations 19
Migrating SETGEN Information. 19
Tables 20
Linkage Types. 20

Chapter 3. CSP/AE 3.2.1 or Earlier to VisualAge Generator Server for MVS, VSE, and VM or VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX 21

Chapter 2. CSP/AD 3.2.1 or Earlier to VisualAge Generator Developer

This chapter lists the considerations for migrating from CSP/AD 3.2.1 or earlier releases to VisualAge Generator Developer.

To migrate from CSP/AD 3.2.1 or earlier releases to VisualAge Generator, first migrate your MSLs to VisualAge Generator and save the external source file used for migration.

MSLs

If you are migrating from CSP/AD 3.2.1 or earlier releases, you need to complete the following steps to convert the Member Specification Library (MSL) members:

1. Export, in internal source format, the entire MSL from CSP/AD 3.2.1 or earlier.
2. Import the internal source format for the entire MSL into CSP/AD 3.2.2, CSP/AD 3.3, or CSP/370AD 4.1. These are the versions of Cross System Product that support both internal and external source format.
3. Export, in external source format, the entire MSL from CSP/AD 3.2.2, CSP/AD 3.3, or CSP/370AD 4.1.

Note: Be sure to save the exported external source format for use in migrating resource association information to VisualAge Generator.

4. Download the export file to the workstation.
5. Use the VisualAge Generator Migration Assistance Tool to migrate the export file into the VisualAge Generator library manager. Refer to the *VisualAge Generator Guide to Migrating MSLs to ENVY* for information about using the Migration Assistance Tool.

Cross System Product MSL members are converted to VAGen parts during this process.

Note: Steps 2, 3 and 5 above can be accomplished through a service offering.

Considerations for Migrating an MSL from VSE/ESA to the Workstation

After you have converted your CSP/AD 3.2.1 or earlier MSLs to CSP/AD 3.3, you must create an external source format file to transfer to the workstation.

The JCL shown in Figure 1 on page 15 provides a method of exporting your MSL to a VSE library member where it can be easily transferred to the workstation. First, the entire MSL or selected MSL member is exported to tape as an external source format file. The DITTO executable file is used to write a /* following the external source format file to signify end-of-data to LIBR, the VSE librarian program. The LIBR program is run to catalog the external source format file as a VSE library member. The sublibrary and member name information are passed to the LIBR program on the PARM statement. After successful completion of this job, the newly cataloged member can be transferred directly to the workstation. The member can then be imported into a pseudo-MSL using the VisualAge Generator Migration Assistance Tool.

Notes:

1. Prior to running this sample JCL, please ensure that APAR PN53933 has been applied to CSP/AE 3.3 (program number 5668-814).
2. Refer to the *VisualAge Generator Guide to Migrating MSLs to ENVY* for information about using the Migration Assistance Tool.


```

* $$ JOB JNM=MOVEMSL,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB MOVEMSL MOVE CSP330 MSL TO TAPE
// DLBL USERCAT, 'USER.CATALOG',,VSAM
// DLBL DCAMAPD, 'CSP.V3R3M0.FZEMAPDS',,VSAM,CAT=USERCAT
// DLBL DCATESD, 'CSP.V3R3M0.FZETUTOR',,VSAM,CAT=USERCAT
// DLBL DCAEZED, 'CSP.V3R3M0.EZEMSG',,VSAM,CAT=USERCAT
// DLBL DCAHECD, 'CSP.V3R3M0.FZEMSG',,VSAM,CAT=USERCAT
// DLBL TESTMSL, 'CSP330.SAMPMSL',,VSAM,CAT=USERCAT
// MTC REW,281
// ASSGN SYS006,281
// LIBDEF *,SEARCH=(PRD2.DBASE)
// ASSGN SYS005,SYSLST
// EXEC DCBYINIT,SIZE=64K
ROMSL=(TESTMSL) CMDIN=SYSIPT CMDOUT=EZEPRINT
F=ESFOUT L=NO S=006 B=80 R=NO
F=EZEPRINT S=005
;;
EXPORT TYPE(ALL) SERIAL(ESFOUT) FORMAT(EXTERNALU);
/*
// UPSI 1
// EXEC DITTO
$$DITTO BSR OUTPUT=281,NBLKS=1
$$DITTO CT OUTPUT=281,BLKFACTOR=1
/+
/*
$$ DITTO EOJ
/*
// MTC REW,281
// ASSGN SYSIPT,281
// EXEC LIBR,PARM='ACC S=PRD2.TEMPMSL;CATALOG TEMPMSL.Z REPLACE=YES'
/*
/&
* $$ EOJ

```

Figure 1. Sample VSE/ESA JCL for Migrating CSP/AD 3.2.1 Members

To avoid the need for tape handling in exporting members, you can use a VSAM entry sequenced data set (ESDS) as an external source format file. Start by defining a VSAM ESDS with a fixed-length size of 80 bytes as shown in Figure 2 on page 16 and define a CICS FCT entry as shown in Figure 3 on page 16. Then, use the JCL shown in Figure 1 to export the entire MSL or selected MSL members to the VSAM ESDS shown in Figure 4 on page 17. After successful completion of this job, the data set can be transferred directly to the workstation.

VSAM ESDS Delete/Define Job

```
* $$ JOB JNM=CWCIDCAM, CLASS=A, DISP=D, NTFY=YES
* $$ LST FND=1STK, CLASS=A, DISP=H
// JOB DEFINE TEST003
* $$ PRT FNO=1STK, CLASS=A, REMOTE=0, DISP=H
// EXEC IDCAMS, SIZE=AUTO
DELETE (TEST003.VSAMESDS.TESTFILE.CSPCAT ) -
    CLUSTER -
    PURGE -
    CATALOG (CSP.VSAM.CATALOG)
DEFINE CLUSTER ( -
    NAME (TEST003.VSAMESDS.TESTFILE.CSPCAT) -
    TRACKS (200 5) 0
    SHAREOPTIONS (2) -
    RECORDSIZE (80 6000) -
    VOLUMES (VOL706) -
    REUSE -
    NONINDEXED -
    FREESPACE (15 7) -
    TO (99366)) -
    DATA (NAME (TEST003.VSAMESDS.TESTFILE.CSPCAT.@d@)) -
    CATALOG (CSP.VSAM.CATALOG)

/*
/&
* $$ EOJ
```

Figure 2. VSAM ESDS DELETE/DEFINE JOB Example

CICS FCT Entry

```
DFHFCT TYPE=DATASET,
        DATASET=TEST003,
        ACCMETH=(VSAM, ESDS) , LSRPOOL=NONE,
        SERVREQ=(READ, UPDATE, ADD, DELETE, BROWSE),
        BUFND=6,
        BUFNI=3,
        STRNO=3,
        FILSTAT=(DISABLED, CLOSED)
```

Figure 3. CICS FCT Entry Example

CSP 3.3 Export Job

```
* $$ JOB JNM=CSPXPORT, CLASS=1
* $$ LST FNO=1STK, CLASS=Q, DISP=H
// JOB CSPXPORT
// ASSGN SYS005, SYSLST
// DLBL DCAMAPD, 'CSP.V3R3M0.FZEMAPDS', , VSAM, CAT=CSPCAT
// DLBL DCAEZED, 'CSP.V3R3M0.EZEMSG', , VSAM, CAT=CSPCAT
// DLBL DCAHECD, 'CSP.V3R3M0.FZEMSG', , VSAM, CAT=CSPCAT
// DLBL DCATESD, 'CSP.V3R3M0.FZETUTOR', , VSAM, CAT=CSPCAT
// DLBL FZERSAM, 'CSP.USERSAM.ALF', , VSAM, CAT=CSPCAT
// DLBL DCAWORK, 'CSP.USER.WORK', , VSAM, CAT=CSPCAT
*
// DLBL TEST003, 'TEST003.VSAMESDS.TESTFILE.CSPCAT', , VSAM, CAT=CSPCAT
*
// DLBL TBL1MSL, 'CSP.TBL1MSL.MSL', , VSAM, CAT=CSPCAT
// DLBL MAP1MSL, 'CSP.MSL1MSL.MSL', , VSAM, CAT=CSPCAT
// DLBL REC1MSL, 'CSP.REC1MSL.MSL', , VSAM, CAT=CSPCAT
// DLBL USR1MSL, 'CSP.USR1MSL.MSL', , VSAM, CAT=CSPCAT
// DLBL USR2MSL, 'CSP.USR2MSL.MSL', , VSAM, CAT=CSPCAT
// DLBL USR3MSL, 'CSP.USR3MSL.MSL', , VSAM, CAT=CSPCAT
// DLBL USR4MSL, 'CSP.USR4MSL.MSL', , VSAM, CAT=CSPCAT
// EXEC DCBYINIT, SIZE=64K
M=USER3MSL ROMSL= (TBL1MSL, REC1MSL, MAP1MSL) CDMIN=SYSIPT CMDOUT=EZEPRINT
EXPORT MEMBER (LAPPL3) SERIAL (TEST003) TYPE (ASSOCIATES) FORMAT (EXTERNAL);
/*
/&
* $$ E0J
```

Figure 4. CSP 3.3 EXPORT JOB Example

Message File Conversion

See “Appendix F. Using the Message File Conversion Utility” on page 123 for more information on using the conversion utility.

Test Facility Limitations

The VisualAge Generator test facility emulates the /MATH=COBOL method of storing intermediate results for arithmetic calculations. The VisualAge Generator test facility does not emulate the Cross System Product method /MATH=CSPAE.

Testing TSO or CICS Programs

If your Cross System Product applications call non-Cross System Product programs, you need to plan for testing this scenario. Some techniques that you might use, depending on your old Cross System Product environment and what the non-VisualAge Generator program does, are as follows:

- Use a stub VisualAge Generator program to simulate what the non-VisualAge Generator program does so that all testing can occur on the workstation.
- Create a workstation version of the non-VisualAge Generator program. The workstation version can be a C or COBOL program in any of the following environments:
 - CICS OS/2
 - CICS Transaction Server for AIX
 - CICS Transaction Server for NT
 - OS/2
 - Windows NT
- Use a remote call to the host version of the non-VisualAge Generator program. This might require using the PARMFORM=COMMPTR setting in the linkage table to call the host.

Note: While testing, you can only call non-Cross System Product programs that are running in a CICS environment. TSO non-Cross System Product programs cannot be called from the workstation during testing.

Definition Limitations

Cross System Product has a limit of 4096 rows in a table. You can define larger tables with VisualAge Generator.

VisualAge Generator does not support the internal source format that was supported by Cross System Product.

See “Chapter 3. CSP/AE 3.2.1 or Earlier to VisualAge Generator Server for MVS, VSE, and VM or VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX” on page 21 for considerations that might apply to development and generation.

Special Function Words

EZECNVCM is the same as EZEDLTRM. Setting one of the language elements sets the other. EZEDLTRM support is only for compatibility with releases of the Cross System Product set prior to 3.3.

Generation Options

In CSP/AD some generation options were saved in the application MSL member. In VisualAge Generator, generation information is not saved in the program part. If you want to use the same generation options for more than one generation, specify the options in generation control parts such as the generation options part and the resource association part.

Refer to the *VisualAge Generator Generation Guide* for information about generation options.

Resource Associations

In CSP/AD, information associating application record definitions with physical files in the target environment was saved in the application MSL member from one generation to the next. With VisualAge Generator, if you want to use the same file resource association information across more than one generation, you must specify the information in a resource association part.

Resource association information saved in application MSL members can be converted for VisualAge Generator by doing the following:

1. Export external source format from CSP/AD 3.2.2 or later.
2. Review the resulting external source format file to get the attribute values specified on the :GENFILE tag. Use that resource information as input, and create resource association parts. If the SYSTEM attribute is not specified on the :GENFILE tag, use the SYSTEM value specified on the :TARGSYS tag instead. See the *VisualAge Generator Generation Guide* for the format of the resource association part.

Note: The resource association part used by VisualAge Generator Developer is not compatible with the resource association file used by the Interactive Test Facility (ITF). Therefore, you need to create resource association parts for use in generation and a resource association file for use by the ITF.

Migrating SETGEN Information

In CSP/AD the SETGEN batch command enabled the developer to choose which map groups and tables to generate with an application. The SETGEN command also enabled the application developer to set table-related options, such as SHARED, RESIDENT, and KEEP AFTER USE, during generation. VisualAge Generator provides the following:

- The SETGEN command has been removed.
- The /GENMAPS and /GENHELPMAPS generation options are used to replace the SETGEN command's ability to specify which map groups to generate.
- The /GENTABLES generation option is used to specify that all the tables associated with the program are to be generated. You can no longer selectively choose which tables to generate.
- The SHARED, RESIDENT, and KEEP AFTER USE options have been moved to VisualAge Generator as definition attributes. The SHARED and RESIDENT options are specified in table definition. The KEEP AFTER USE option is specified during 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. The change for SHARED and RESIDENT options therefore affects all programs that contain the respective tables.

Tables

The generation options RESIDENT, SHARED, and KEEP AFTER USE were generation options in CSP/AD. They are specified at definition time with VisualAge Generator. SHARED and RESIDENT are specified within table definition. KEEP AFTER USE is specified on the Tables and Additional Record window in program definition.

Linkage Types

- **CICS**
For Cross System Product releases prior to 4.1, the compatible linkage type and parameter format for CICS environments is CICS LINK COMMPTR.
- **non-CICS**
For Cross System Product releases prior to 4.1, the compatible linkage type and parameter format for non-CICS environments is DYNAMIC OSLINK.

Chapter 3. CSP/AE 3.2.1 or Earlier to VisualAge Generator Server for MVS, VSE, and VM or VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX

See “Appendix B. Cross System Product Interpretive to COBOL” on page 83 for considerations when migrating to a generated COBOL environment. See “Appendix C. Cross System Product Interpretive to C++” on page 111 for considerations when migrating to a generated C++ environment. See “Appendix D. Host Environments to VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX” on page 119 if you are changing from a host environment to an OS/2, AIX, Windows, or HP-UX environment.

Part 3. Migrating from Cross System Product 3.2.2 or 3.3

Chapter 4. CSP/AD 3.2.2 or CSP/AD 3.3 to VisualAge Generator Developer	25	Templates	40
MSLs.	25	Generation Output Libraries on the Host	42
Considerations for Migrating an MSL from VSE/ESA to the Workstation	25	Interfacing to PL/I	43
Message File Conversion	28	SQL Plan Definition	43
Test Facility Limitations	28	Chapter 8. CSP/370RS 1.1 to VisualAge Generator Server for MVS, VSE, and VM	45
Testing TSO or CICS Programs	28	Installation Considerations	45
Definition Limitations	29	Procedures	45
Special Function Words	30	Upward Compatibility	46
Generation Options	30	Generating Applications, Tables, and Map Groups Again	46
Resource Associations	30	Generated Tables	47
Migrating SETGEN Information	31	Applications Accessing the Same Table	47
Tables	31	Error Routines	47
Linkage Types.	31	Defining PSBs	47
Chapter 5. CSP/AD 3.3 Programmable Workstation Feature to VisualAge Generator Developer	33	SQL Plan Definition	48
CSP/AD 3.3 Programmable Workstation Feature MSLs	33	User Messages	48
Definition Limitations	33		
Batch Commands	34		
NLS Parameters	34		
Chapter 6. CSP/AE 3.2.2 or 3.3 to VisualAge Generator Server for MVS, VSE, and VM or VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX	35		
Chapter 7. CSP/370RS COBOL Generation to VisualAge Generator Developer	37		
Generation Options	37		
Generation Options in the MSL	37		
Generation Options File	37		
Resource Associations	38		
Migrating SETGEN Information	39		
Tables	39		
Linkage Types.	40		
Command Interface	40		
Reserved Word File	40		

Chapter 4. CSP/AD 3.2.2 or CSP/AD 3.3 to VisualAge Generator Developer

This chapter lists the considerations for migrating from CSP/AD 3.2.2 or CSP/AD 3.3 to VisualAge Generator Developer.

To migrate from CSP/AD to VisualAge Generator Developer, first migrate your MLSs to VisualAge Generator and save the external source file used for migration.

MSLs

If you are migrating from either CSP/AD 3.2.2 or CSP/AD 3.3, you need to complete the following steps to convert the Member Specification Library (MSL) members:

1. Export, in external source format, the entire MSL from CSP/AD 3.2.2 or CSP/AD 3.3.

Note: Be sure to save the exported external source format for use in migrating resource association information to VisualAge Generator.

2. Download the export file to the workstation.
3. Use the VisualAge Generator Migration Assistance Tool to migrate the export file into the VisualAge Generator library manager. Refer to the *VisualAge Generator Guide to Migrating MSLs to ENVY* for information about using the Migration Assistance Tool.

CSP/AD 3.2.2 and CSP/AD 3.3 MSL members are converted to VAGen parts during this process.

Note: Step 3 above can be accomplished through a service offering.

Considerations for Migrating an MSL from VSE/ESA to the Workstation

The JCL shown in Figure 5 on page 26 provides a method of exporting your MSL to a VSE library member where it can be easily transferred to the workstation. First, the entire MSL or selected MSL member is exported to tape as an external source format file. The DITTO executable file is used to write a /* following the external source format file to signify end-of-data to LIBR, the VSE librarian program. The LIBR program is run to catalog the external source format file as a VSE library member. The sublibrary and member name information are passed to the LIBR program on the PARM statement. After

successful completion of this job, the newly cataloged member can be transferred directly to the workstation. The member can then be imported into a pseudo-MSL using the VisualAge Generator Developer Migration Assistance Tool.

Notes:

1. Prior to running this sample JCL, please ensure that APAR PN53933 has been applied to CSP/AE 3.3 (program number 5668-814).
2. Refer to the *VisualAge Generator Guide to Migrating MSLs to ENVY* for information about using the Migration Assistance Tool.

```
* $$ JOB JNM=MOVEMSL,CLASS=A,DISP=D
* $$ LST CLASS=A,DISP=D
// JOB MOVEMSL MOVE CSP330 MSL TO TAPE
// DLBL USERCAT, 'USER.CATALOG',,VSAM
// DLBL DCAMAPD,'CSP.V3R3M0.FZEMAPDS',,VSAM,CAT=USERCAT
// DLBL DCATESD,'CSP.V3R3M0.FZETUTOR',,VSAM,CAT=USERCAT
// DLBL DCAEZED,'CSP.V3R3M0.EZEMSG',,VSAM,CAT=USERCAT
// DLBL DCAHECD,'CSP.V3R3M0.FZEMSG',,VSAM,CAT=USERCAT
// DLBL TESTMSL,'CSP330.SAMPMSL',,VSAM,CAT=USERCAT
// MTC REW,281
// ASSGN SYS006,281
// LIBDEF *,SEARCH=(PRD2.DBASE)
// ASSGN SYS005,SYSLST
// EXEC DCBYINIT,SIZE=64K
ROMSL=(TESTMSL) CMDIN=SYSIPT CMDOUT=EZEPRINT
F=ESFOUT L=NO S=006 B=80 R=NO
F=EZEPRINT S=005
;;
EXPORT TYPE(ALL) SERIAL(ESFOUT) FORMAT(EXTERNALU);
/*
// UPSI 1
// EXEC DITTO
$$DITTO BSR OUTPUT=281,NBLKS=1
$$DITTO CT OUTPUT=281,BLKFACTOR=1
/+
/*
$$ DITTO EOJ
/*
// MTC REW,281
// ASSGN SYSIPT,281
// EXEC LIBR,PARM='ACC S=PRD2.TEMPMSL;CATALOG TEMPMSL.Z REPLACE=YES'
/*
/&
* $$ EOJ
```

Figure 5. Sample VSE/ESA JCL for Migrating CSP/AD 3.3 Members

To avoid the need for tape handling in exporting members, you can use a VSAM entry sequenced data set (ESDS) as an external source format file. Start by defining a VSAM ESDS with a fixed-length size of 80 bytes as shown in

Figure 6 and define a CICS FCT entry as show in Figure 7 . Then, use the JCL shown in Figure 1 on page 15 to export the entire MSL or selected MSL members to the VSAM ESDS shown in Figure 8 on page 28. After successful completion of this job, the data set can be transferred directly to the workstation.

VSAM ESDS Delete/Define Job

```
* $$ JOB JNM=CWCIDCAM, CLASS=A, DISP=D, NTFY=YES
* $$ LST FND=1STK, CLASS=A, DISP=H
// JOB DEFINE TEST003
* $$ PRT FNO=1STK, CLASS=A, REMOTE=0, DISP=H
// EXEC IDCAMS, SIZE=AUTO
DELETE (TEST003.VSAMESDS.TESTFILE.CSPCAT ) -
    CLUSTER -
    PURGE -
    CATALOG (CSP.VSAM.CATALOG)
DEFINE CLUSTER ( -
    NAME (TEST003.VSAMESDS.TESTFILE.CSPCAT) -
    TRACKS (200 5) 0
    SHAREOPTIONS (2) -
    RECORDSIZE (80 6000) -
    VOLUMES (VOL706) -
    REUSE -
    NONINDEXED -
    FREESPACE (15 7) -
    TO (99366)) -
    DATA (NAME (TEST003.VSAMESDS.TESTFILE.CSPCAT.@d@) -
    CATALOG (CSP.VSAM.CATALOG)

/*
/&
* $$ EOJ
```

Figure 6. VSAM ESDS DELETE/DEFINE JOB Example

CICS FCT Entry

```
DFHFCT TYPE=DATASET,
    DATASET=TEST003,
    ACCMETH=(VSAM, ESDS) , LSRPOOL=NONE,
    SERVREQ=(READ, UPDATE, ADD, DELETE, BROWSE),
    BUFND=6,
    BUFNI=3,
    STRNO=3,
    FILSTAT=(DISABLED, CLOSED)
```

Figure 7. CICS FCT Entry Example

CSP 3.3 Export Job

```
* $$ JOB JNM=CSPXPORT, CLASS=1
* $$ LST FNO=1STK, CLASS=Q, DISP=H
// JOB CSPXPORT
// ASSGN SYS005, SYSLST
// DLBL DCAMAPD, 'CSP.V3R3M0.FZEMAPDS', , VSAM, CAT=CSPCAT
// DLBL DCAEZED, 'CSP.V3R3M0.EZEMSG', , VSAM, CAT=CSPCAT
// DLBL DCAHECD, 'CSP.V3R3M0.FZEMSG', , VSAM, CAT=CSPCAT
// DLBL DCATESD, 'CSP.V3R3M0.FZETUTOR', , VSAM, CAT=CSPCAT
// DLBL FZERSAM, 'CSP.USERSAM.ALF', , VSAM,CAT=CSPCAT
// DLBL DCAWORK, 'CSP.USER.WORK', , VSAM, CAT=CSPCAT
*
// DLBL TEST003, 'TEST003.VSAMESDS.TESTFILE.CSPCAT', , VSAM, CAT=CSPCAT
*
// DLBL TBL1MSL, 'CSP.TBL1MSL.MSL', , VSAM, CAT=CSPCAT
// DLBL MAP1MSL, 'CSP.MSL1MSL.MSL', , VSAM, CAT=CSPCAT
// DLBL REC1MSL, 'CSP.REC1MSL.MSL', , VSAM, CAT=CSPCAT
// DLBL USR1MSL, 'CSP.USR1MSL.MSL', , VSAM, CAT=CSPCAT
// DLBL USR2MSL, 'CSP.USR2MSL.MSL', , VSAM, CAT=CSPCAT
// DLBL USR3MSL, 'CSP.USR3MSL.MSL', , VSAM, CAT=CSPCAT
// DLBL USR4MSL, 'CSP.USR4MSL.MSL', , VSAM, CAT=CSPCAT
// EXEC DCBYINIT, SIZE=64K
M=USER3MSL ROMSL= (TBL1MSL, REC1MSL, MAP1MSL) CDMIN=SYSIPT CMDOUT=EZEPRINT
EXPORT MEMBER (LAPPL3) SERIAL (TEST003) TYPE (ASSOCIATES) FORMAT (EXTERNAL);
/*
/&
* $$ E0J
```

Figure 8. CSP 3.3 EXPORT JOB Example

Message File Conversion

See “Appendix F. Using the Message File Conversion Utility” on page 123 for more information on using the conversion utility.

Test Facility Limitations

The VisualAge Generator test facility emulates the /MATH=COBOL method of storing intermediate results for arithmetic calculations. The VisualAge Generator test facility does not emulate the Cross System Product method /MATH=CSPAE.

Testing TSO or CICS Programs

If your Cross System Product applications call non-Cross System Product programs, you need to plan for testing this scenario. Some techniques that you might use, depending on your old Cross System Product environment and what the non-VisualAge Generator program does, are as follows:

- Use a stub VisualAge Generator program to simulate what the non-VisualAge Generator program does so that all testing can occur on the workstation.
- Create a workstation version of the non-VisualAge Generator program. The workstation version can be a C or COBOL program in any of the following environments:
 - CICS OS/2
 - CICS Transaction Server for AIX
 - CICS Transaction Server for NT
 - OS/2
 - Windows NT
- Use a remote call to the host version of the non-VisualAge Generator program. This might require using the PARMFORM=COMMPTR setting in the linkage table to call the host.

Note: While testing, you can only call non-Cross System Product programs that are running in a CICS environment. TSO non-Cross System Product programs cannot be called from the workstation during testing.

Definition Limitations

Cross System Product has a limit of 4096 rows in a table. You can define larger tables with VisualAge Generator.

The external source format for VisualAge Generator is not compatible with the external source format for CSP/AD 3.2.2 or 3.3. You can import the external source format from CSP/AD 3.2.2 or 3.3 into the VisualAge Generator Developer Migration Assistance Tool. However, you cannot import the external source format from VisualAge Generator Developer back into CSP/AD 3.2.2 or 3.3.

See the following for considerations that might apply to development and generation:

- “Chapter 6. CSP/AE 3.2.2 or 3.3 to VisualAge Generator Server for MVS, VSE, and VM or VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX” on page 35
- “Chapter 7. CSP/370RS COBOL Generation to VisualAge Generator Developer” on page 37
- “Chapter 8. CSP/370RS 1.1 to VisualAge Generator Server for MVS, VSE, and VM” on page 45

Special Function Words

EZECNVCM is the same as EZEDLTRM. Setting one of the language elements sets the other. EZEDLTRM support is only for compatibility with releases of the Cross System Product set prior to 3.3.

Generation Options

In CSP/AD some generation options were saved in the application MSL member. In VisualAge Generator, generation information is not saved in the program part. If you want to use the same generation options for more than one generation, specify the options in generation control parts, such as the generation options part and the resource association part.

Refer to the *VisualAge Generator Generation Guide* for information about generation options.

Resource Associations

In CSP/AD, information associating application record definitions with physical files in the target environment was saved in the application MSL member from one generation to the next. With VisualAge Generator, if you want to use the same file resource association information across more than one generation, you must specify the information in a resource association part.

Resource association information saved in application MSL members can be converted for VisualAge Generator by doing the following:

- Export external source format from CSP/AD.
- Review the resulting external source format file to get the attribute values specified on the :GENFILE tag. Use that resource information as input, and create resource association parts. From the **File** menu in the Parts Editor window, select the **Read from File** option to retrieve the resource association information created in step 1. If the SYSTEM attribute is not specified on the :GENFILE tag, use the SYSTEM value specified on the :TARGSYS tag instead.

Note: The resource association part used by VisualAge Generator during generation is not compatible with the resource association file used by the Interactive Test Facility (ITF). Therefore, you need to create resource association parts for use in generation and a resource association file for use by the ITF.

Migrating SETGEN Information

In CSP/AD the SETGEN batch command enabled the developer to choose which map groups and tables to generate with an application. The SETGEN command also enabled the application developer to set table related options, such as SHARED, RESIDENT, and KEEP AFTER USE, during generation. VisualAge Generator provides the following:

- The SETGEN command has been removed.
- The /GENMAPS and /GENHELPMAPS generation options are used to replace the SETGEN command's ability to specify which map groups to generate.
- The /GENTABLES generation option is used to specify that all the tables associated with the program are to be generated. You can no longer selectively choose which tables to generate.
- The SHARED, RESIDENT, and KEEP AFTER USE options have been moved to VisualAge Generator as definition attributes. The SHARED and RESIDENT options are specified in table definition. The KEEP AFTER USE option is specified during 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. The change for SHARED and RESIDENT options therefore affects all programs that contain the respective tables.

Tables

The generation options RESIDENT, SHARED, and KEEP AFTER USE were generation options in CSP/AD. They are specified at definition time with VisualAge Generator. SHARED and RESIDENT are specified within table definition. KEEP AFTER USE is specified on the Tables and Additional Record window in program definition.

Linkage Types

- **CICS**
For Cross System Product releases prior to 4.1, the compatible linkage type and parameter format for CICS environments is CICSLINK COMMPTR.
- **non-CICS**
For Cross System Product releases prior to 4.1, the compatible linkage type and parameter format for non-CICS environments is DYNAMIC OSLINK.

Chapter 5. CSP/AD 3.3 Programmable Workstation Feature to VisualAge Generator Developer

This chapter lists the considerations for migrating from CSP/AD 3.3 programmable workstation feature to VisualAge Generator Developer.

VisualAge Generator can coexist on the same system as CSP/AD 3.3 Programmable Workstation Feature; you can run both products concurrently. Refer to the documentation for more information on startup parameters.

CSP/AD 3.3 Programmable Workstation Feature MSLs

If you are migrating from CSP/AD 3.3 Programmable Workstation Feature, you need to complete the following steps to convert the Member Specification Library (MSL) members:

1. Export, in external source format, the entire MSL from CSP/AD 3.3 Programmable Workstation Feature.

Note: Be sure to save the exported external source format for use in migrating resource association information to VisualAge Generator.

2. Use the VisualAge Generator Developer Migration Assistance Tool to migrate the export file into the VisualAge Generator library manager. Refer to the *VisualAge Generator Guide to Migrating MSLs to ENVY* for information about using the Migration Assistance Tool.

CSP/AD 3.3 Programmable Workstation Feature MSL members are converted to VAGen parts during this process.

Note: Step 2 above can be accomplished through a service offering.

Definition Limitations

The external source format for VisualAge Generator is not compatible with the external source format for CSP/AD 3.3 Programmable Workstation Feature. You can import the external source format from CSP/AD 3.3 Programmable Workstation Feature into the VisualAge Generator Developer Migration Assistance Tool. However, you cannot import the external source format from VisualAge Generator Developer back into CSP/AD 3.3 Programmable Workstation Feature.

See the following for considerations that might apply to development and generation:

- “Chapter 4. CSP/AD 3.2.2 or CSP/AD 3.3 to VisualAge Generator Developer” on page 25
- “Chapter 6. CSP/AE 3.2.2 or 3.3 to VisualAge Generator Server for MVS, VSE, and VM or VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX” on page 35
- “Chapter 7. CSP/370RS COBOL Generation to VisualAge Generator Developer” on page 37
- “Chapter 8. CSP/370RS 1.1 to VisualAge Generator Server for MVS, VSE, and VM” on page 45

Batch Commands

CSP/AD 3.3 Programmable Workstation Feature batch command files are compatible with VisualAge Generator, with these exceptions:

- Commands related to MSLs are not supported in VisualAge Generator because MSLs are no longer used.
- The command syntax has changed for those commands that are still available in VisualAge Generator. See the *VisualAge Generator Programmer’s Reference* for the correct command syntax.

NLS Parameters

In CSP/AD 3.3, the syntax for the NLS parameter was a 1-character NLS code. In VisualAge Generator, the syntax for the NLS parameter has changed to a 3-character NLS code.

NLS language codes changed as shown in the following table:

Table 3. Language Support Suffixes

Description	V3.3	VisualAge Generator
U.S. English (Mixed case)	E	ENU
Simplified Chinese	C	CHS
German (Germany)	G	DEU
German (Switzerland)	W	DES
Japanese	J	JPN
Korean	K	KOR
Portuguese (Brazilian)	P	PTB
Spanish	S	ESP

Chapter 6. CSP/AE 3.2.2 or 3.3 to VisualAge Generator Server for MVS, VSE, and VM or VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX

See “Appendix B. Cross System Product Interpretive to COBOL” on page 83 for considerations when migrating to a generated COBOL environment. See “Appendix C. Cross System Product Interpretive to C++” on page 111 for considerations when migrating to a generated C++ environment. See “Appendix D. Host Environments to VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX” on page 119 if you are changing from a host environment to an OS/2, AIX, Windows, or HP-UX environment.

Chapter 7. CSP/370RS COBOL Generation to VisualAge Generator Developer

CSP/AD 3.3 required CSP/370RS 1.1 to generate COBOL programs. This section lists the considerations for migrating from COBOL generation using CSP/370RS 1.1 to COBOL generation using VisualAge Generator Developer.

To migrate from CSP/370RS 1.1 COBOL Generation to the VisualAge Generator Developer, first migrate your MSLs to VisualAge Generator and save the external source format (ESF) file used for migration. See “MSLs” on page 25 for information on how to migrate your MSLs.

Generation Options

In CSP/370RS, some generation options were saved in the application MSL member and some options were specified in COBOL generation options files.

Generation Options in the MSL

In VisualAge Generator, generation information is not saved in the program part. If you want to use the same generation options for more than one generation, you must specify the options in generation control parts, such as the generation options part and the resource association part.

Generation Options File

The COBOL generation options file was accessed as a sequential file (ddname EKZGOPT) in CSP/370RS 1.1 COBOL generation. Override situations were handled using concatenated files. For example, an installation options file was allocated to the EKZGOPT ddname. Project and user options files were concatenated after the installation options file. The last value encountered for a generation option was the option that was used, so the user options overrode the installation options.

VisualAge Generator Developer also uses generation options; however, they have a different format. Override situations are handled using the /OPTIONS keyword to point to a lower priority generation options part. Because the first value encountered for a generation option is the option that is used, the /OPTIONS keyword on the GENERATE subcommand should specify the user options part.

In addition, VisualAge Generator Developer enables you to use a generation options default part where you can specify installation defaults and prevent them from being overridden. You can specify a default generation options part in the hpt.ini file under the defaultGenerationOptions parameter.

Note: VisualAge Generator uses the default file name EFKOPxxx.OPT for the options file, where xxx varies based on the target environment. The 8775-1C, 2C, 3C, and 4C devices are not valid in the /MFSDEV generation option.

Generation options used in CSP/370RS 1.1 can be migrated for use with VisualAge Generator Developer. You must first convert the CSP/370RS generation options files that you want to use into VisualAge Generator Developer generation options parts. The format for these generation options parts is similar to that for other workstation products and thus differs from the format of the CSP/370RS generation options files.

Follow these steps to do the conversion:

1. Review your CSP/370RS generation options. See “Appendix G. Host Generation Option Changes” on page 129 for more information on the relationship between CSP/370RS generation options and the corresponding VisualAge Generator generation options.
2. Review the chapter on generation commands and options in the *VisualAge Generator Generation Guide* to determine if you need to specify any new generation options.
3. Based on the information collected in steps 1 and 2, create one or more generation options parts in VisualAge Generator to contain your generation options.

Resource Associations

In CSP/AD 3.3 with CSP/370RS 1.1 COBOL generation, information associating application record definitions with physical files in the target environment was saved in the application MSL member from one generation to the next. With VisualAge Generator, if you want to use the same file resource association information across more than one generation, you must specify the information in a resource association part.

Resource association information saved in application MSL members can be converted for VisualAge Generator by doing the following:

1. Export external source format from CSP/AD.
2. Review the resulting external source format file to get the attribute values specified on the :GENFILE tag. Use that resource information as input, and create resource association parts. From the **File** menu in the Parts Editor

window, select the **Read from File** option to retrieve the resource association information created in step 1. If the SYSTEM attribute is not specified on the :GENFILE tag, use the SYSTEM value specified on the :TARGSYS tag instead.

Note: The resource association part used by VisualAge Generator Developer during generation is not compatible with the resource association file used by the Interactive Test Facility (ITF). Therefore, you need to create resource association parts for use in generation and a resource association file for use by the ITF.

Migrating SETGEN Information

In CSP/AD with CSP/370RS COBOL generation, the SETGEN batch command enabled the developer to choose which map groups and tables to generate with an application. The SETGEN command also enabled the application developer to set table related options, such as SHARED, RESIDENT, and KEEP AFTER USE, during generation. VisualAge Generator provides the following:

- The SETGEN command has been removed.
- The /GENMAPS and /GENHELPMAPS generation options are used to replace the SETGEN command's ability to specify which map groups to generate.
- The /GENTABLES generation option is used to specify that all the tables associated with the program are to be generated. You can no longer selectively choose which tables to generate.
- The SHARED, RESIDENT, and KEEP AFTER USE options have been moved to VisualAge Generator as definition attributes. The SHARED and RESIDENT options are specified in table definition. The KEEP AFTER USE option is specified during 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. The change for SHARED and RESIDENT options therefore affects all programs that contain the respective tables.

Tables

The generation options RESIDENT, SHARED, and KEEP AFTER USE were generation options in CSP/370RS 1.1 COBOL generation. They are specified at definition time with VisualAge Generator. SHARED and RESIDENT are specified within table definition. KEEP AFTER USE is specified on the Tables and Additional Record window in program definition.

Linkage Types

The compatible linkage type is DYNAMIC OSLINK.

Command Interface

The CSP/AD 3.3 with CSP/370RS 1.1 COBOL generation command interface is similar to the format used for the VisualAge Generator Developer commands. If you have CSP/AD 3.3 commands that you want to use, you must convert them to the workstation format. Utilities are not provided to do this conversion.

Reserved Word File

The default reserved words for VisualAge Generator Developer are kept in the EFKRSV.RSV file. This file identifies reserved names that are assigned aliases in generated COBOL applications. The file contains some additional words in VisualAge Generator Developer. If you modified the reserved words in the file specified by the EKZRESVD DD statement (CSP/370RS 1.1), you need to make the same modifications in the EFKRSV.RSV file. Refer to the *VisualAge Generator Generation Guide* for information on modifying the reserved word file.

Templates

Different templates (model JCL members for generating preparation and execution JCL) are shipped with VisualAge Generator Developer. The VisualAge Generator Developer templates might require customization to fit your installation requirements for JCL usage and data set names. The CSP/370RS 1.1 templates cannot be used with VisualAge Generator Developer because the names were changed and templates were combined. See Table 4 for a list of the CSP/370RS 1.1 template names and the corresponding VisualAge Generator Developer names.

Table 4. CSP/370RS 1.1 Template Names and Corresponding VisualAge Generator Developer Template Names

CSP/370RS 1.1 Template Names	VisualAge Generator Developer Template Names
EKZMMCL	EFK2MPBA
EKZMMPCL	EFK2MPBC
EKZMMCLB	None, messages are in tables
EKZDMCL	EFK2MPBA

Table 4. CSP/370RS 1.1 Template Names and Corresponding VisualAge Generator Developer Template Names (continued)

CSP/370RS 1.1 Template Names	VisualAge Generator Developer Template Names
EKZDMPCL	EFK2MPBB
EKZDMCLB	None, messages are in tables
EKZBMCL	EFK2MPIA
EKZBMPCL	EFK2MPIB
EKZBMCLB	None, messages are in tables
EKZIMCL	EFK2MPIC
EKZIMPCL	EFK2MPIE
EKZIMCLB	None, messages are in tables
EKZITCL	EFK2MPIC
EKZITPCL	EFK2MPIE
EKZITCLB	EFK2MPID
EKZMCCL	EFK2MPBA
EKZMCPCL	EFK2MPBC
EKZMCCLB	None, messages are in tables
EKZDCCL	EFK2MPBA
EKZDCPCL	EFK2MPBB
EKZDCCLB	None, messages are in tables
EKZBCCL	EFK2MPIA
EKZBCPCL	EFK2MPIB
EKZBCCLB	None, messages are in tables
EKZICCL	EFK2MPIC
EKZIPCL	EFK2MPIE
EKZICCLB	None, messages are in tables
EKZBIND	EFK2MBDB, EFK2MBDA, or EFK2MBDD, depending on the target environment and database usage
EKZBINDR	EFK2MBDC
EKZMSP	EFK2MMCB for MFS mapping services program or EFK2MMCA for batch mapping services program
EKZMFSU	EFK2MMSU
EKZMFST	EFK2MMST

Table 4. CSP/370RS 1.1 Template Names and Corresponding VisualAge Generator Developer Template Names (continued)

CSP/370RS 1.1 Template Names	VisualAge Generator Developer Template Names
EKZTBL	EFK2MMCA
EKZBMP	EFK2MEIB
EKZBMP2	EFK2MEIA
EKZMDLI	EFK2MEBC
EKZMDB2	EFK2MEBD
EKZMDBS	EFK2MEBB
EKZMBACH	EFK2MEBE
EKZCALL	EFK2MEBA
EKZQSDDI	EFK2MSDI
EKZQSDDO	EFK2MSDO
EKZVSDDI	EFK2MVSII
EKZVSDDO	EFK2MVSO
EKZGSDDI	EFK2MGSI
EKZGSDDO	EFK2MGSO
EKZMDBDD	EFK2MDLI
EKZMDGSM	EFK2MIMS
EKZUMSDD	None, messages are in tables
EKZMMSDB	None, messages are in tables

Generation Output Libraries on the Host

VisualAge Generator Developer uses different library naming conventions. An environment qualifier has been added as the middle-level qualifier of the data set names for generation output libraries. Refer to the *VisualAge Generator Generation Guide* for a description of how to set up libraries for generation for VisualAge Generator Developer.

Interfacing to PL/I

Static calls to PL/I programs are supported by VisualAge Generator. However, you must generate the program with VisualAge Generator, using a linkage table if you want to call PL/I programs directly. Refer to the *VisualAge Generator Client/Server Communications Guide* for a description of how to set up a linkage table.

SQL Plan Definition

If any SQL applications in a run unit were generated on CSP/370RS to use a DB2 message database, then include the database request module (DBRM) ELADBRM2 in the plan associated with the run unit.

Chapter 8. CSP/370RS 1.1 to VisualAge Generator Server for MVS, VSE, and VM

CSP/AD 3.3 required CSP/370RS 1.1 to generate COBOL programs. The following sections list the considerations for migrating CSP/370RS 1.1 using generated COBOL programs to using VisualAge Generator Server for MVS, VSE, and VM.

You can use VisualAge Generator Server for MVS, VSE, and VM to replace CSP/370RS 1.1 as a runtime environment. You can modify any JCL (batch, IMS, CICS, or MVS/TSO) or CLIST that allocated the CSP/370RS 1.1 load library to use the VisualAge Generator Server for MVS, VSE, and VM load library.

Installation Considerations

VisualAge Generator Server for MVS, VSE, and VM must be installed in a separate SMP/E zone and have different target libraries from CSP/370RS 1.1. VisualAge Generator Server for MVS, VSE, and VM does not include any of the COBOL generation function that was included in CSP/370RS 1.1, so do not delete CSP/370RS 1.1 from your system until you have migrated both the COBOL generation and runtime services functions to VisualAge Generator.

If you placed any CSP/370RS 1.1 load modules in the LPA, replace them with the VisualAge Generator Server for MVS, VSE, and VM load modules before migration is complete. If you removed the CSP/370RS load modules from the SELALMD load library when you placed the load modules in the LPA, you must put them back in the SELALMD load library when the load modules are removed from the LPA. Otherwise, any JCL or CLISTs that allocate the CSP/370RS 1.1 load library will cause unpredictable results. The code being run will be a combination of CSP/370RS 1.1 code from the load library and VisualAge Generator Server for MVS, VSE, and VM code from the LPA.

Procedures

The preparation procedures are shipped in VisualAge Generator Server for MVS, VSE, and VM. The procedure names are not changed from CSP/370RS 1.1 to VisualAge Generator Server for MVS, VSE, and VM.

The cataloged procedures for program preparation shipped with VisualAge Generator Server for MVS, VSE, and VM work only with preparation JCL

generated on the workstation using the VisualAge Generator generation facility. The VisualAge Generator Server for MVS, VSE, and VM procedures do not work with the preparation JCL generated using CSP/370RS 1.1. If you plan to use CSP/370RS 1.1 for generation, keep a copy of the CSP/370RS 1.1 cataloged procedures for use with applications generated with CSP/370RS 1.1.

Upward Compatibility

Generating Applications, Tables, and Map Groups Again

In general, applications, tables, and map groups generated with CSP/370RS 1.1 do not have to be generated again to use them with VisualAge Generator Server for MVS, VSE, and VM. The VisualAge Generator generation facility supports unnamed map variables in a map group. In addition, the VisualAge Generator generation facility eliminates the mapping services program for terminal maps in all environments and adds a map group format module for the IMS environments. The generation facility also adds information to tables that can be used as edit routines, such as match valid, match invalid, and range match valid tables. The ability to share tables and map groups between an application generated with CSP/370RS 1.1 and a program generated with the VisualAge Generator generation facility depends on the following:

- Whether the generation was done by CSP/370RS 1.1 or VisualAge Generator Developer
- Whether the map group has any unnamed map variables

The rules for using tables and map groups are as follows:

- If you generate a program with VisualAge Generator, you must also generate any edit tables and map groups with VisualAge Generator that the program uses.
- If you generate the map groups for the program, you must also generate any other programs with VisualAge Generator that use the map groups.
- If you generate a map group with VisualAge Generator, you must also generate all tables that are specified as edit routines for any map variable field in the map group with VisualAge Generator. Any programs that use the map group must be generated with VisualAge Generator.
- If the map group contains any unnamed map variables, the map group and all programs that use the map group must be generated with VisualAge Generator.
- If you generate a table with VisualAge Generator, you do not need to generate any programs or map groups with VisualAge Generator that use the table.

Notes:

1. As soon as any program, table, or map group is generated with VisualAge Generator, VisualAge Generator Server for MVS, VSE, and VM must be used.
2. If you do not follow these rules, you might receive the messages ELA00046P, ELA00051P, or ELA00208P, or ABEND code ELAW in MVS CICS or ABEND code 1610 in non-CICS environments. Refer to the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information on these messages and ABEND codes.

Generated Tables

Tables generated under CSP/370RS 1.1 might have used AMODE(24). If you use these tables under MVS CICS, you must link them again as AMODE(31). If you use these tables for non-CICS environments, you can use AMODE(24), but AMODE(31) is recommended.

Applications Accessing the Same Table

Running a generated CSP/370RS 1.1 application with a generated VisualAge Generator program in the same run unit where both access the same table can cause an abend or wrong data to be retrieved from the table. CSP/370RS 1.1 applications and VisualAge Generator generated programs handle tables differently. To avoid problems, define the table as resident.

Error Routines

If EZEFEC is set to 1 and there is an error routine specified, VisualAge Generator Server for MVS, VSE, and VM returns control to the program on OPEN and CLOSE errors. CSP/370RS 1.1 does not. The program does not have to be generated again to use this function.

Defining PSBs

The ELAPCB macro for VisualAge Generator Server for MVS, VSE, and VM supports only the work database. PSBs for applications generated using CSP/370RS 1.1 might also have used a message database. If you generate the PSB for these applications again, you must use the ELAPCB macro from CSP/370RS 1.1.

If you used a DL/I message database with CSP/370RS 1.1, you included ELAMSG as a PCB in the PSB. ELAMSG is supported by VisualAge Generator Server for MVS, VSE, and VM for compatibility with CSP/370RS 1.1.

SQL Plan Definition

If any SQL applications in a run unit were generated on CSP/370RS to use a DB2 message database, then include the database request module (DBRM) ELADBRM2 in the plan associated with the run unit.

User Messages

The message utility is not shipped with VisualAge Generator. User messages are no longer in VSAM files, DL/I databases, or DB2 tables. Instead, they are in VisualAge Generator tables. Applications generated under CSP/370RS 1.1 to use a message file or database are supported by VisualAge Generator. However, if you need to change the text of a message in the message database, you will need to use the CSP/370RS 1.1 message utility.

Part 4. Migrating from Cross System Product 4.1

Chapter 9. CSP/370AD 4.1 to VisualAge Generator Developer	51
MSLs	51
Message File Conversion	51
Test Facility Limitations	52
Testing TSO or CICS Programs	52
Definition Limitations	52
Generation Options	53
Generation Options in the MSL	53
Generation Options File	53
Resource Associations	54
Migrating SETGEN Information	54
Tables	55
Linkage Tables	55
Command Interface	55
Reserved Word File	55
Templates	56
Generation Output Libraries on the Host	58
Chapter 10. CSP/2AD 1.2 to VisualAge Generator Developer	59
CSP/2AD 1.2 MSLs	59
Defintion Limitations	59
Batch Commands	59
NLS Parameters	59
Chapter 11. CSP/370RS 2.1 to VisualAge Generator Server for MVS, VSE, and VM	61
Installation Considerations	61
Procedures	62
Upward Compatibility	62
Generating Applications, Tables, and Map Groups Again	62
Chapter 12. CSP/2RS 1.1 to VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX	65
CSP/2RS File Migration Considerations	65
Upward Compatibility	67
OS/2 CICS User Data Files	68
Chapter 13. Running CSP/370AD 4.1 or VisualAge Generator 2.2 Applications with VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX 3.0 or Later.	69

Chapter 9. CSP/370AD 4.1 to VisualAge Generator Developer

This chapter lists the considerations for migrating from CSP/370AD 4.1 to VisualAge Generator Developer.

To migrate from CSP/370AD 4.1 to VisualAge Generator Developer, first migrate your MSLs to VisualAge Generator and save the external source file used for migration.

MSLs

If you are migrating from CSP/370AD 4.1, you need to complete the following steps to convert the Member Specification Library (MSL) members:

1. Export, in external source format, the entire MSL from CSP/370AD 4.1.

Note: Be sure to save the exported external source format for use in migrating resource association information to VisualAge Generator Developer.

2. Download the export file to the workstation.
3. Use the VisualAge Generator Migration Assistance Tool to migrate the export file into the VisualAge Generator library manager. Refer to the *VisualAge Generator Guide to Migrating MSLs to ENVY* for information about using the Migration Assistance Tool.

Cross System Product MSL members are converted to VAGen parts during this process.

Note: Step 3 above can be accomplished through a service offering.

Message File Conversion

See “Appendix F. Using the Message File Conversion Utility” on page 123 for more information on the conversion utility.

Test Facility Limitations

The VisualAge Generator test facility emulates the /MATH=COBOL method of storing intermediate results for arithmetic calculations. The VisualAge Generator test facility does not emulate the Cross System Product method /MATH=CSPAЕ.

Testing TSO or CICS Programs

If your Cross System Product applications call non-Cross System Product programs, you need to plan for testing this scenario. Depending on your old Cross System Product environment and what the non-VisualAge Generator program does, techniques that you might use include:

- Use a stub VisualAge Generator program to simulate what the non-VisualAge Generator program does so that all testing can occur on the workstation.
- Create a workstation version of the non-VisualAge Generator program. The workstation version can be a C or COBOL program in any of the following environments:
 - CICS OS/2
 - CICS Transaction Server for AIX
 - CICS Transaction Server for NT
 - OS/2
 - Windows NT
- Use a remote call to the host version of the non-VisualAge Generator program. This might require using the PARMFORM=COMMPTR setting in the linkage table to call the host.

Note: While testing, you can only call non-Cross System Product programs that are running in a CICS environment. TSO non-Cross System Product programs cannot be called from the workstation during testing.

Definition Limitations

Cross System Product has a limit of 4096 rows in a table. You can define larger tables with VisualAge Generator.

The external source format for VisualAge Generator is not compatible with the external source format for CSP/370AD 4.1. You can import the external source format from CSP/370AD 4.1 into the VisualAge Generator Developer Migration Assistance Tool. However, you cannot import the external source format from VisualAge Generator Developer back into CSP/370AD 4.1.

See “Chapter 11. CSP/370RS 2.1 to VisualAge Generator Server for MVS, VSE, and VM” on page 61 for considerations that might also apply to development and generation.

Generation Options

In CSP/370AD 4.1, some generation options were saved in the application MSL member and some options were specified in COBOL generation options files.

Generation Options in the MSL

In VisualAge Generator, generation information is not saved in the program part. If you want to use the same generation options for more than one generation, you must specify the options in generation control parts, such as the generation options part and the resource association part.

Generation Options File

Generation options that were specified in a generation options file used in CSP/370AD 4.1 can be migrated for use with VisualAge Generator Developer. You must convert any CSP/370AD 4.1 generation options files that you want to use to VisualAge Generator generation options parts. The format for these generation options parts is similar to that for other workstation products and thus differs from the format of the CSP/370AD 4.1 generation option files.

Follow these steps to do the conversion:

1. Review your CSP/370AD 4.1 generation options. See “Appendix G. Host Generation Option Changes” on page 129 for more information on the relationship between CSP/370AD 4.1 generation options and the corresponding VisualAge Generator generation options.
2. Review the chapter on generation commands and options in the *VisualAge Generator Generation Guide* to determine if you need to specify any new generation options.
3. Based on the information collected in steps 1 and 2, create one or more generation options parts in VisualAge Generator to contain your generation options.

Resource Associations

In CSP/370AD 4.1, information associating application record definitions with physical files in the target environment was saved in the application MSL member from one generation to the next. With VisualAge Generator, if you want to use the same file resource association information across more than one generation, you must specify the information in a resource association part.

CSP/370AD 4.1 resource association information saved in application MSL members can be converted for VisualAge Generator by doing the following:

1. Export external source format from CSP/370AD.
2. Review the resulting external source format file to get the attribute values specified on the :GENFILE tag. Use that resource information as input, and create resource association parts. If the SYSTEM attribute is not specified on the :GENFILE tag, use the SYSTEM value specified on the :TARGSYS tag instead. See the *VisualAge Generator Generation Guide* for the format of the resource association part.

Note: The resource association part used by the VisualAge Generator Developer is not compatible with the resource association file used by the Interactive Test Facility (ITF). Therefore, you need to create resource association parts for use in generation and a resource association file for use by the ITF.

Migrating SETGEN Information

In CSP/370AD the SETGEN batch command enabled the developer to choose which map groups and tables to generate with an application. The SETGEN command also enabled the application developer to set table related options, such as SHARED, RESIDENT, and KEEP AFTER USE, during generation. VisualAge Generator Developer provides the following:

- The SETGEN command has been removed.
- The /GENMAPS and /GENHELPMAPS generation options are used to replace the SETGEN command's ability to specify which map groups to generate.
- The /GENTABLES generation option is used to specify that all the tables associated with the program are to be generated. You can no longer selectively choose which tables to generate.
- The SHARED, RESIDENT, and KEEP AFTER USE options have been moved to VisualAge Generator as definition attributes. The SHARED and RESIDENT options are specified in table definition. The KEEP AFTER USE option is specified during 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. The change for SHARED and RESIDENT options therefore affects all programs that contain the respective tables.

Tables

The generation options RESIDENT, SHARED, and KEEP AFTER USE were generation options in CSP/370AD 4.1. They are specified at definition time with VisualAge Generator. SHARED and RESIDENT are specified within table definition. KEEP AFTER USE is specified on the Tables and Additional Record window in program definition.

Linkage Tables

In CSP/370AD, linkage tables were stored as separate files. With VisualAge Generator, this information is stored in a linkage table part. You can migrate your linkage tables by doing the following:

1. Download the CSP/370AD linkage table file to the workstation.
2. Create a linkage table part in VisualAge Generator. Then from the **File** menu in the **Parts Editor** window, select **Read from File** to retrieve your old linkage table.

Command Interface

The CSP/370AD 4.1 command interface is similar to the format used for the VisualAge Generator Developer commands. If you have CSP/370AD commands that you want to use, you must convert them to the workstation format. Utilities are not provided to do this conversion.

Reserved Word File

No migration is necessary for the reserved word file. You can transfer the file from the host and use it without changes. The default reserved word file for VisualAge Generator Developer is EFKRSV.RSV.

The reserved word file for CSP/370AD 4.1 was the file specified by the EZEWORD DD statement in your generation JCL.

Templates

The template names are different between CSP/370AD and VisualAge Generator. The templates are now located in the \template subdirectory under the VisualAge Generator Developer product directory.

Any templates and procedures that you modified must be modified again when migrating from CSP/370AD to the VisualAge Generator application generation facility.

The contents of the templates used by VisualAge Generator Developer might be different from CSP/370AD, so you should not use the CSP/370AD generation templates with VisualAge Generator Developer. If you customized your templates and want to make the same changes to the VisualAge Generator Developer templates, see Table 5 for the CSP/370AD template names and the corresponding VisualAge Generator Developer template names. The procedure names have not changed.

Table 5. CSP/370AD Template Names and the Corresponding VisualAge Generator Developer JCL Template Names

CSP/370AD Template Names	VisualAge Generator Developer Template Names
EZEACL	EFK2MMCA
EZEBCL	EFK2MPIA
EZEBIND	EFK2MBDB
EZEBINDA	EFK2MBDA
EZEBINDN	EFK2MBDD
EZEBINDR	EFK2MBDC
EZEBMP	EFK2MEIB
EZEBMP2	EFK2MEIA
EZEBPCLB	EFK2MPIB
EZECALL	EFK2MEBA
EZECCL	EFK2MPCB
EZECPCLB	EFK2MPCA
EZEDPCLB	EFK2MPBB
EZEGSDDI	EFK2MGSI
EZEGSDDO	EFK2MGSO
EZEICL	EFK2MPIC
EZEICLB	EFK2MPID

Table 5. CSP/370AD Template Names and the Corresponding VisualAge Generator Developer JCL Template Names (continued)

CSP/370AD Template Names	VisualAge Generator Developer Template Names
EZEIPCLB	EFK2MPIE
EZELINKR	EFK2MPRE
EZEMBACH	EFK2MEBE
EZEMCL	EFK2MPBA
EZEMDB2	EFK2MEBD
EZEMDBAL	EFK2MTDL
EZEMDBDD	EFK2MDLI
EZEMDBS	EFK2MEBB
EZEMDEST	EFK2MEZD
EZEMDGSM	EFK2MIMS
EZEMDLI	EFK2MEBC
EZEMFACL	EFK2MMCB
EZEMFST	EFK2MMST
EZEMFSU	EFK2MMSU
EZEMPCLB	EFK2MPBC
EZEMXAPP	EFK2MCAL
EZEMXEAP	EFK2MEZA
EZEQSALI	EFK2MTSI
EZEQSALO	EFK2MTSO
EZESQDDI	EFK2MSDI
EZEQSDDO	EFK2MSDO
EZETCALL	EFK2META
EZETCL	EFK2MPTA
EZETDB2	EFK2METC
EZETDEST	EFK2MTED
EZETDLI	EFK2METB
EZETPCLB	EFK2MPTB
EZETSO	EFK2METD
EZETXAPP	EFK2MTCL
EZETXEAP	EFK2MTEA
EZEVSALI	EFK2MTVI

Table 5. CSP/370AD Template Names and the Corresponding VisualAge Generator Developer JCL Template Names (continued)

CSP/370AD Template Names	VisualAge Generator Developer Template Names
EZEVSALO	EFK2MTVO
EZEVSDDI	EFK2MVSJ
EZEVSDDO	EFK2MVSO

Generation Output Libraries on the Host

When you generate for an MVS target environment, the generation outputs created by VisualAge Generator Developer are transferred to data sets that use the same default naming conventions that were used by CSP/370AD 4.1.

Specify /CICSENTRIES=RDO if you want to generate model resource definition online (RDO) program and transaction definitions. Specify /CICSENTRIES=MACRO if you want to generate model PPT and PCT table entries.

Chapter 10. CSP/2AD 1.2 to VisualAge Generator Developer

VisualAge Generator can coexist on the same system as CSP/2AD 1.2; you can run both products concurrently. Refer to the documentation for more information on startup parameters.

CSP/2AD 1.2 MSLs

The VisualAge Generator Migration Assistance Tool can access CSP/2AD 1.2 MSLs. Refer to *VisualAge Generator Guide to Migrating MSLs to ENVY* for information about loading the VisualAge Generator library manager.

Defintion Limitations

The external source format for VisualAge Generator is not compatible with the external source format for CSP/2AD 1.2. You can import the external source format from CSP/2AD 1.2 into the VisualAge Generator Developer Migration Assistance Tool. However, you cannot import the external source format from VisualAge Generator Developer back into CSP/2AD 1.2.

See “Chapter 9. CSP/370AD 4.1 to VisualAge Generator Developer” on page 51 and “Chapter 11. CSP/370RS 2.1 to VisualAge Generator Server for MVS, VSE, and VM” on page 61 for considerations that might apply to development and generation.

Batch Commands

CSP/2AD 1.2 batch command files are compatible with VisualAge Generator. However, commands related to MSLs are not supported in VisualAge Generator because MSLs are no longer used.

NLS Parameters

The syntax for the NLS parameter for CSP/2AD 1.2 is compatible with VisualAge Generator except that the languages supported have been modified. Refer to the *VisualAge Generator Installation Guide* for more information on the NLS parameters.

Chapter 11. CSP/370RS 2.1 to VisualAge Generator Server for MVS, VSE, and VM

You can use VisualAge Generator Server for MVS, VSE, and VM to replace CSP/370RS 2.1 as a runtime environment. You can modify any JCL (batch, IMS, CICS, or MVS/TSO) or CLISTs that allocated the CSP/370RS 2.1 load library to use the VisualAge Generator Server for MVS, VSE, and VM load library.

Installation Considerations

VisualAge Generator Server for MVS, VSE, and VM must be installed in a separate SMP/E zone and have different target libraries from CSP/370RS 2.1 and CSP/370AD 4.1. If you do not have CSP/370AD 4.1 installed on your system, you can delete CSP/370RS 2.1 after you migrate all your generated applications to the VisualAge Generator Server for MVS, VSE, and VM runtime environment. If you have both CSP/370AD 4.1 and CSP/370RS 2.1 installed on your system, you cannot delete CSP/370RS 2.1 until you do the following:

- Migrate all your generated applications and programs to use the VisualAge Generator Server for MVS, VSE, and VM runtime environment
- Migrate all your development and generation to VisualAge Generator Developer.

Once you have completed the above procedures, you can then delete both CSP/370AD 4.1 and CSP/370RS 2.1.

If you placed any CSP/370RS 2.1 load modules in the LPA, replace them with the VisualAge Generator Server for MVS, VSE, and VM load modules before migration is complete. If you removed the CSP/370RS load modules from the SELALMD load library when you placed the load modules in the LPA, you must put them back in the SELALMD load library when the load modules are removed from the LPA. Otherwise, any JCL or CLISTs that allocate the CSP/370RS 2.1 load library will cause unpredictable results. The code being run will be a combination of CSP/370RS 2.1 code from the load library and VisualAge Generator Server for MVS, VSE, and VM code from the LPA.

Procedures

The procedure names for program preparation have not changed from CSP/370RS 2.1 to VisualAge Generator Server for MVS, VSE, and VM. The cataloged procedures for program preparation shipped with VisualAge Generator Server for MVS, VSE, and VM work only with preparation JCL generated on the workstation using VisualAge Generator Developer. The VisualAge Generator Server for MVS, VSE, and VM procedures do not work with the preparation JCL generated using CSP/370AD 4.1. If you plan to use CSP/370AD 4.1 for development and generation, keep a copy of the CSP/370RS 2.1 cataloged procedures for use with applications generated with CSP/370AD 4.1.

Upward Compatibility

Generating Applications, Tables, and Map Groups Again

In general, applications, tables, and map groups generated with CSP/370AD 4.1 do not have to be generated again to use them with VisualAge Generator Server for MVS, VSE, and VM. The VisualAge Generator generation facility eliminates the mapping services program for terminal maps in all environments and adds a map group format module for the IMS environments. The generation facility also adds information to tables that can be used as edit routines, such as match valid, match invalid, and range match valid tables. The ability to share tables and map groups between an application generated with CSP/370AD 4.1 and a program generated with the VisualAge Generator generation facility depends on the following:

- Whether the generation was done by CSP/370AD 4.1 or VisualAge Generator Developer

The rules for using tables and map groups are as follows:

- If you generate a program with VisualAge Generator, you must also generate any edit tables and map groups with VisualAge Generator that the program uses.
- If you generate the map groups for the program, you must also generate any other programs with VisualAge Generator that use the map groups.
- If you generate a map group with VisualAge Generator, you must also generate all tables that are specified as edit routines for any map variable field in the map group with VisualAge Generator. Any programs that use the map group must be generated with VisualAge Generator.
- If you generate a table with VisualAge Generator, you do not need to generate any programs or map groups with VisualAge Generator that use the table.

Notes:

1. As soon as any program, table, or map group is generated with VisualAge Generator, then VisualAge Generator Server for MVS, VSE, and VM must be used.
2. If you do not follow these rules, you might receive the messages ELA00046P, ELA00051P, or ELA00208P, or ABEND code ELAW in MVS CICS or ABEND code 1610 in non-CICS environments. Refer to the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information on these messages and ABEND codes.

Chapter 12. CSP/2RS 1.1 to VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX

This chapter lists considerations for migrating from CSP/2RS 1.1 to VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX in the CICS OS/2 environment.

CSP/2RS and VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX cannot be used concurrently on the same machine. CSP/2RS should be removed from the machine before VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX is installed. If this is not practical, remove all references to CSP/2RS from the CONFIG.SYS while you are using VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX.

If you customized any of the files associated with CSP/2RS 1.1, you need to perform similar customizations on VisualAge Generator command files and templates.

CSP/2RS File Migration Considerations

ELARUNC.CMD

This file is provided with VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX to start CICS OS/2 with VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX support. However, the file has been significantly modified. ELARUNC.CMD calls ELAENV.CMD to establish the runtime environment. Most modifications are placed in the ELAENV.CMD file. Review versions of the ELARUNC.CMD and ELAENV.CMD files supplied with VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX to determine if any customizations need to be migrated.

ELAPRSET.BAK

The ELAPRSET.CMD file (shipped as ELAPRSET.BAK) is not provided with VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX. Make runtime modifications to the workstation environment in the ELAENV.CMD file. If any customizations were made to ELAPRSET.CMD, review ELAENV.CMD to determine if any of these customizations need to be migrated.

ELARXCIC.CMD, ELARXCOB.CMD, ELARXLNK.CMD, and ELAPRCTL.CTL

If you customized these files for CSP/2RS and intend to use the VisualAge Generator generation facility, you might need to migrate your customizations to the templates supplied with VisualAge

Generator Developer. Specifically, determine if the templates EFK2OPXC.TPL, EFK2OPXF.TPL, and EFK2OPXP.TPL require any customization.

For more information on these templates, refer to the *VisualAge Generator Generation Guide*.

If you will be using the preparation utility for CSP/370AD members, you might need to migrate your customizations to the versions of these files provided with VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX.

ELACNxxx.DLL

Data conversion files that were customized can be used with VisualAge Generator.

ELAxxx.ESF and ELACxxx.ESF

If you customized these files for CSP/2RS and intend to use VisualAge Generator Developer, you might need to migrate your customizations to the programs supplied with VisualAge Generator Developer.

Note: CSP/2RS 1.1 supported distributing customized files by uploading them to the host and overwriting the CSP/2RS installation files on the host. VisualAge Generator does not support this method of distribution.

During CSP/2RS installation, users added CICS OS/2 table entries under the application group ELA110. VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX adds several entries with the same name under the application group VGSERV. If both application groups exist on the same CICS OS/2 system, CICS OS/2 uses the entries associated with VGSERV because it is listed after ELA110 alphabetically.

If you want to remove the duplicate entries from your system, use the CEDA transaction to delete them. See Table 6 for the duplicate entries:

Table 6. Table of Duplicate Entries

Table	Entry	Group Name
DCT	ELAD	ELA110
FCT	ELACFIL	ELA110
PCT	ELAC	ELA110
PCT	ELAM	ELA110
PCT	ELAN	ELA110
PCT	ELAU	ELA110
PCT	ELAZ	ELA110

Upward Compatibility

With VisualAge Generator Server 3.0 and later, Micro Focus COBOL is no longer supported. Only applications generated into VisualAge for COBOL for OS/2 (also called IBM COBOL) can be used with VisualAge Generator Server 3.0 and later. Therefore, if you previously used Micro Focus COBOL to prepare your applications, you must regenerate all applications, tables, and map groups and then prepare them using IBM COBOL.

The VisualAge Generator generation facility eliminates the mapping services program for terminal maps. The generation facility also adds information to tables that can be used as edit routines, such as match valid, match invalid, and range match valid tables. The ability to share tables and map groups between an application generated with CSP/370AD 4.1 and a program generated with the VisualAge Generator generation facility depends on the following:

- Whether the generation was done by CSP/370AD 4.1 or VisualAge Generator Developer

The rules for using tables and map groups are as follows:

- If you generate a program with VisualAge Generator, you must also generate any edit tables and map groups with VisualAge Generator that the program uses.
- If you generate the map groups for the program, you must also generate any other programs with VisualAge Generator that use the map groups.
- If you generate a map group with VisualAge Generator, you must also generate all tables that are specified as edit routines for any map variable field in the map group with VisualAge Generator. Any programs that use the map group must be generated with VisualAge Generator.
- If you generate a table with VisualAge Generator, you do not need to generate any programs or map groups with VisualAge Generator that use the table.

Notes:

1. As soon as any program, table, or map group is generated with VisualAge Generator, VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX must be used.
2. If you do not follow these rules, you might receive the messages ELA00046P, ELA00051P, or ELA00208P, or receive ABEND code ELAW in the CICS environments. Refer to the chapter about running COBOL applications on OS/2 in the *VisualAge Generator Messages and Problem Determination Guide* for more information on this ABEND code.

OS/2 CICS User Data Files

CICS controlled files (with filetype = VSAM) that were created by a Cross System Product 4.1 generated application can be used by VisualAge Generator programs. COBOL controlled files (with filetype = OS2COBOL) are not compatible when migrating from Micro Focus COBOL to IBM COBOL. For example, an OS2COBOL file created by a Micro Focus COBOL generated application cannot be used with an IBM COBOL generated program. To migrate a Micro Focus COBOL created file, you need to write a Cross System Product 4.1 application that reads one record at a time from the Micro Focus COBOL file and writes it to a CICS VSAM file. CICS VSAM files are compatible with VisualAge Generator programs.

Chapter 13. Running CSP/370AD 4.1 or VisualAge Generator 2.2 Applications with VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX 3.0 or Later

With VisualAge Generator Server 3.0 and later, Micro Focus COBOL is no longer supported. Only applications generated into IBM COBOL can be used with VisualAge Generator Server 3.0 and later. Therefore, if you previously used Micro Focus COBOL to prepare your applications, you must regenerate all applications, tables, and map groups and then prepare them using IBM COBOL.

CICS controlled files (with filetype = VSAM) that were created by a Cross System Product 4.1 generated application can be used by VisualAge Generator programs. COBOL controlled files (with filetype = OS2COBOL) are not compatible when migrating from Micro Focus COBOL to IBM COBOL. For example, an OS2COBOL file created by a Micro Focus COBOL generated application cannot be used with an IBM COBOL generated program. To migrate a Micro Focus COBOL created file, you need to write a Cross System Product 4.1 application that reads one record at a time from the Micro Focus COBOL file and writes it to a CICS VSAM file. CICS VSAM files are compatible with VisualAge Generator programs.

Part 5. Appendixes

Appendix A. Language Differences from the Cross System Product Set

The following sections list the language differences from CSP/370AD 4.1 to VisualAge Generator 3.1, from CSP/AD 3.3 to VisualAge Generator 3.1, and from CSP/AD 3.2.2 to VisualAge Generator 3.1.

Language Differences from CSP/370AD 4.1 to VisualAge Generator

The following sections describe the language differences from CSP/370AD 4.1 to VisualAge Generator.

New and Enhanced Function Words

- The EZECONCT option for CICS OS/2 special function word provides access to the DB2/2 CONNECT, CONNECT TO, and CONNECT RESET functions.
- The CALL keyword is not required for EZE function calls.
- The EZESYS function word now allow you to specify ITF as a system environment.

New String-Handling Function Words

The following new EZE words for string-handling functions are available:

EZESBLKT

Changes null terminator and any subsequent characters in a string to blanks

EZESCCSW

Concatenates one string to another, with a separator string between them

EZESCMPR

Compares one substring to another

EZESCNCT

Concatenates one string to another

EZESCOPY

Copies one substring to another

EZESFIND

Finds the first occurrence of a specified string within a string

EZESNULT

Changes trailing blanks to nulls in a string

EZESSET

Sets each character in a substring to the same character value

EZESTLEN

Returns the length of an item less trailing blanks and nulls

EZESTOKN

Finds the next token in a string and copies the token to an item

New Math Function Words

Some EZE words were added for general math routines, floating-point math routines, and trigonometric math routines.

General Math Routines**EZEABS**

Absolute value

EZECEIL

Smallest integer not less than the numericDataItem

EZEEXP

Exponential value (e raised to the power of numericDataItem)

EZEFLOOR

Largest integer not greater than numericDataItem

EZEFREXP

Split numericDataItem into normalized fraction in range of $\frac{1}{2}$ to 1 and a power of 2

EZELDEXP

Product of numericDataItem multiplied by 2 to the power of integer

EZELOG

Natural logarithm

EZELOG10

Base 10 logarithm

EZEMAX

Maximum

EZEMIN

Minimum

EZEMODF

Split into integral and fractional parts

EZENCMPR

Numeric comparison

EZEPOW

Raise to power

EZEPRCSN

Maximum precision in decimal digits

EZEROUND

Round to integer power of 10

EZESQRT

Square root

Floating-Point Math Routines**EZEFLADD**

Floating point add

EZEFLDIV

Floating point division

EZEFLMOD

Floating point remainder of division

EZEFLMUL

Floating point multiplication

EZEFLSET

Conversion to floating point

EZEFLSUB

Floating point subtraction

Trigonometric Math Routines**EZEACOS**

Arccosine

EZEASIN

Arcsine

EZEATAN

Arctangent

EZEATAN2

Theta component of the polar coordinate corresponding to the rectangular coordinate

EZECOS

Cosine

EZECOSH

Hyperbolic cosine

EZESIN

Sine

EZESINH

Hyperbolic sine

EZETAN

Tangent

EZETANH

Hyperbolic tangent

Client/Server Processing Enhancements

VisualAge Generator now supports client/server processing between the following:

- An MVS CICS program on a host system and CICS OS/2 program on the programmable workstation
- Any IMS and VM programs on any host systems
- Any CICS programs on the same or different host or workstation systems
- GUI clients and any server through use of the CALL statement
- 16-bit and 32-bit DLLs

VisualAge Generator also supports:

- Use of the linkage table from the ITF
- Calls to the ITF from other products

See the *VisualAge Generator Client/Server Communications Guide* for more information on client/server processing.

Cursor with Hold Option

The CURSOR WITH HOLD option specifies whether the DECLARE CURSOR statement issued for a SETINQ or SETUPD process option includes a WITH HOLD clause.

Maximums Increased

- The maximum length for Numeric or PACK fields in SQL row records has been changed from 15 digits to 18 digits. This limit also applies to PACK fields in other types of records.
- Line length has increased from 71 to 2000 characters.

Table Specification Changes

The table specifications KEEP AFTER USE, SHARED, and RESIDENT are now specified at definition, not at generation. SHARED and RESIDENT are

specified within table definition. KEEP AFTER USE is specified on the Tables and Additional Record window in program definition.

Trace Facility Enhancement

The trace facility for generated applications is expanded to trace SQL information and statements.

Host Product Upgrades

VisualAge Generator Server for MVS, VSE, and VM requires LE COBOL. VS COBOL II and SAA COBOL are no longer supported.

CALL Statement Enhancements

The following enhancements are available for the CALL statement:

- Any data item can be passed as a parameter on any CALL.
- Data items can be qualified and subscripted on any CALL.

Language Differences from CSP/AD 3.3 to VisualAge Generator

In addition to the language differences listed in this section, if you are migrating from CSP/AD 3.3 to VisualAge Generator, you should also review the language differences listed under “Language Differences from CSP/370AD 4.1 to VisualAge Generator” on page 73.

The following language enhancements were added to the CSP/370AD and CSP/2AD products, or were added to VisualAge Generator for the VM and VSE environments.

Note: CSP/370AD and CSP/2AD did not support VM or VSE.

Transactional Processing

The following language elements improve support for the transactional processing environments of IMS/VS and MVS CICS and add function for the other supported environments:

- **Special Function Words**

The following special function words supply information about the generated program in the runtime environment:

EZEUSRID	User ID
EZELTERM	Terminal identifier
EZESYS	System environment

- **Longer Records for XFER, DXFR, and CREATX statements**
The records passed on an XFER, DXFR, or CREATX statement can be up to 32KB in length.
- **First Map and XFER with Map**
First map and XFER with map can be specified for programs targeted to run in interactive environments. This support enables you to control the amount of working storage that needs to be saved when a map is displayed. It also permits development and testing of efficient IMS and pseudoconversational CICS transactions to run consistently across all environments.
- **Segmented Program Simulation**
You can specify the execution mode for programs as part of the program specification. In the test facility and the nontransaction runtime environments like MVS/TSO and VM CMS, segmented runtime is simulated by committing recoverable resources and refreshing special function words and single user table contents at each CONVERSE or XFER with map, allowing programs to run consistently across environments.
- **Consistent Commit Points for DXFR and XFER Processing Statements**
Consistent commit processing for DXFR and XFER enables you to control when commit points are performed on DXFR and XFER statements. You use the test facility and COBOL generation options to request consistent commit processing (commit on XFER or no commit on DXFR except when the program specification block changes) across all environments.
- **XFER in Batch Environments**
XFER with record is allowed in batch programs for the MVS batch, IMS BMP, and VM batch environments.

Note: XFER from a called program is not supported in generated COBOL programs or in the test facility.
- **Linkage Table**
A linkage table can be used at generation to select the implementation for a CALL statement, a CREATX statement, file I/O, or a DXFR statement. The linkage table provides a way of specifying when a remote call or file I/O is to be done as well as specifying the parameter format for a CALL statement.

System Resource Association

The following additions for associating records with different types of files provide you with greater flexibility.

- **File Types**
In addition to the file types supported in previous releases of the Cross System Product set, the following file types are supported:

- CICS temporary storage files
- JES Spool files in MVS CICS
- Micro Focus COBOL files on OS/2
- VSE/POWER queue members

Use the resource association part to specify system resource associations.

- **Dynamic File Association and Dynamic File Allocation**

The system resource name associated with a file can be dynamically modified during running of the program for most types of files, using the following special function words:

EZEDEST	Record destination
EZEDESTP	Print destination
EZELOC	Record location. For remote CICS files, the system on which the file is accessed can be modified.

The dynamic file association can be used for the following file types:

- VSAM or sequential files on MVS/TSO, MVS batch, IMS BMP, VSE batch, VM CMS, and VM batch
- IBM COBOL files on CICS OS/2, including standard OS/2 sequential files
- VSAM files, temporary storage queues, transient data queues, and spool files on MVS CICS and VSE CICS
- Message queues on IMS

A generated program dynamically allocates the following types of files for the following environments:

- VSAM or sequential files on MVS/TSO, MVS batch, IMS BMP, VSE batch, VM CMS, and VM batch
- IBM COBOL files on CICS OS/2, including standard OS/2 sequential files

When dynamic allocation is used, the system resource name is the name of the data set or file to be allocated.

Dynamic file association and dynamic file allocation let you build programs that enable the user to identify the data set to be accessed when running the program.

- **Serial ADD and SCAN in Single Program**

A program can include ADD and SCAN processes for the same serial file. The code generated for the program automatically closes and opens the file again when switching from ADD to SCAN.

Client/Server Processing Support

VisualAge Generator supports client/server processing between the following:

- An MVS CICS program on a host system and CICS OS/2 program on the programmable workstation
- MVS CICS programs on different host systems.

The client/server processing support is built on the CALL, CREATX, and file I/O processing options. Using the VisualAge Generator linkage table, communication between applications on different systems can be defined to call a program on a remote system, start an asynchronous transaction on a remote system, or access a file on a remote system.

Processing Statement Extensions

The following processing statement extensions provide you with greater control over processing statements:

- **Parentheses in Arithmetic Statements**

You can use parentheses in arithmetic and conditional statements to control the order in which expressions are evaluated.

- **Sign Support**

A negative (–) or positive (+) sign can be specified for numeric literals, data items, and expressions in parentheses in the arithmetic statement.

- **Data Movement using the Assignment Statement**

You can perform data movement and computation using the assignment statement in the form RESULT=EXPRESSION.

- **Conditional Statements**

The conditional statements, IF and WHILE, can contain parentheses in conditional expressions with the AND and OR operators.

- **IF, WHILE, and TEST Statements**

You can do the following:

- Check the state of a map item by using the keywords CURSOR, DATA, BLANK, or NULL on IF and WHILE
- Use the keyword BLANK with data items or map items on the IF, WHILE, and TEST statements to check whether the contents are blanks
- Use the keyword NUMERIC with map items or data items on the IF, WHILE, and TEST statements to check whether the contents are numeric

- **Blank Lines**

You can enter blank lines; they are stored by statement definition.

Date Edit Mask for Map and Data Item Definition

You can specify a unique date edit mask during map or data item definition. Both 2-digit and 4-digit years are supported. Date edit is supported in both numeric and character fields, and the system date can be retrieved in either numeric or character form. The default date format can be Gregorian or Julian and is specified in the hpt.ini file. You can also select the date format with special keywords.

Set Extended Attributes

The SET statement has been expanded to allow dynamic control for color and extended highlighting attributes of variable map items on display maps.

SQL Dynamic Table Access

SQL table names can be specified as host variables in SQL process options. Dynamic SQL table access allows a table name to be specified in host variable format during the SQL row record definition. The statements for processes that access the record are prepared and run dynamically, substituting the data item value for the table name host variable.

EZECONCT Option

For MVS, the EZECONCT special function word provides access to the DB2 CONNECT, CONNECT TO, and CONNECT RESET functions.

For VSE, the EZECONCT special function word provides access to the DB2/VSE CONNECT and CONNECT TO functions.

For VM, the EZECONCT special function word provides access to the SQL/DS CONNECT function.

Passing of PCBs in EZEDLPCB

An individual PCB can be passed on a CALL statement by subscripting EZEDLPCB with the PCB number to be passed.

User Message Files as Tables

User message files are developed and managed as standard VisualAge Generator tables. This process eliminates any need to separately create and maintain message files outside of VisualAge Generator. In addition, a utility is provided to convert existing VSAM user message files to VisualAge Generator tables. For more information on message file conversion, see “Appendix F. Using the Message File Conversion Utility” on page 123.

Language Differences from CSP/AD 3.2.2 to VisualAge Generator

In addition to the language differences listed in this section, if you are migrating from CSP/AD 3.2.2 or earlier releases to VisualAge Generator, you should also review the language differences listed under “Language Differences from CSP/370AD 4.1 to VisualAge Generator” on page 73 and “Language Differences from CSP/AD 3.3 to VisualAge Generator” on page 77.

The following table lists the language enhancements that were added between CSP/AD 3.2.2 and CSP/AD 3.3.

- Names for data items, records, processes, and statement groups can now be defined as follows:

Table 7. Length of Member Names

Member Name	Length
Data Item	32
Record	18
Process	18
Statement Group	18

- Data items can be local to records or tables. Global data items are defined as separate parts in the library. Local data items are stored only within the record. This allows the same local data item name to be used in different records and have different definitions.
- SQL support was enhanced as follows:
 - SQL multi-column indexes
 - Database switching
 - Concurrent DL/I and DB2 support in MVS batch applications
 - The dynamic WHERE clause
 - Single-row SELECT
 - Modification of FOR UPDATE OF and INSERT INTO clauses in SQL statements built by CSP/AD
 - The ability to generate a model SQL statement for the SQLEXEC process option.

Appendix B. Cross System Product Interpretive to COBOL

This section lists considerations that apply when migrating from CSP/AE to VisualAge Generator Server for MVS, VSE, and VM or VisualGen Host Services for OS/400.

Note: Unless otherwise specified, references in this appendix to VisualAge Generator Server for MVS, VSE, and VM refer to both that product and VisualGen Host Services for OS/400.

Part Names

The following are considerations for part names:

- Program, map group, and table names
 - Must be unique within a CICS runtime system and in a target load library
 - Must not be COBOL reserved words
- Map group names cannot contain the special characters \$, #, and @.
- Care must be exercised to avoid conflicts with derived object names.
Map group format modules are named by appending FM to the map group name. Print mapping services programs for MVS batch, BMP, and VM batch environments are named by appending P1 to the map group name. Therefore, programs and tables should not be named so that they begin with the same characters as a map group name and end with FM or P1.
- To avoid aliases being assigned during COBOL generation for process, statement group, record, data item, map item, and table column names and to improve the readability of the generated COBOL program, use a name that meets the following COBOL naming conventions:
 - Do not use COBOL reserved words
 - Do not use \$, #, @ or _ characters
 - Do not use DBCS member names if the program contains SQL statements or if the CICS translator does not support DBCS names. If the CICS translator does support DBCS names (CICS/ESA Version 3 or later), you must specify the /CICSDBCS generation option to indicate that aliases for DBCS names are not required.

Language Element Compatibility Considerations

The following gives an overview of the compatibility considerations for language elements. The language elements include the following:

- Mapping support
- CONVERSE process option
- EZE special function words
- Transferring among programs
- Programming statements
- File support
- Relational database (SQL) support
- DL/I database support
- Data values
- User message files
- Language-dependent variables
- Language elements associated with the application load files

Mapping Support

Migration considerations for mapping support are as follows:

- If null is specified as the fill character, specify the /NULLFILL generation option to have null values used for CHA, MIX, and DBCS print map variables. Specifying the generation option /NONULLFILL results in blanks being used as the fill character. Using blank instead of null results in the use of more efficient COBOL statements.
- If left justification is specified, specify the /LEFTJUST generation option to have CHA, MIX, and DBCS map variables left-justified on output. Specifying the generation option /NOLEFTJUST causes variables that had JUSTIFY LEFT to be displayed as stored. The /LEFTJUST generation option has no effect on input. If the program always ensures that values are left justified on input, then justification on output is not required, and the generated output editing code is more efficient.
- An automatic form feed for each printer file is issued whenever a main program ends or when a program called by a non-VisualAge Generator program returns to the calling program. An automatic form feed is also issued for the current printer file when a CLOSE process option is explicitly issued. No form feed occurs when a program starts or when a called program returns to another VisualAge Generator program.
- Anabend (such as 0C5) could result from too many maps or too many total fields on maps in a map group.
- Only one definition of a floating area is supported for single-byte printers. If different floating areas are defined for different print devices (for

example, PRINT, PRINT-B, 3767), the definition that is used depends on the order in which the maps were defined and cannot be predicted. To avoid confusion, either specify the same print device for all print maps or specify the same floating area for all single-byte printers.

- All devices that are used for the map must be specified in map definition. For performance reasons, generated COBOL programs cannot display any map on any device; they can only be displayed on the devices specified in map definition.
- The display of a negative zero value is different for CSP/AE and COBOL applications. If a numeric variable field is defined with zero edit and a leading or trailing sign, and a value -0 is moved to the item, CSP/AE displays the negative sign. However, the same map field in the COBOL program will not display the negative sign. If the field is defined with a trailing sign, it will be displayed as $0+$ when running the COBOL program. If the field is defined with a leading sign, it will be displayed as 0 .
- For efficiency, COBOL picture edits are used to implement output edit functions. If there is not enough room in the output field for displaying both the value and edit symbols, COBOL truncates the numeric value instead of deleting edit symbols. Ensure that the field length is long enough to display the largest number intended for the field, in addition to the edit characters.

For example, if the value 123456 was moved to a 6-character map field defined with a currency symbol and a decimal places value of 2, CSP/AE would display the value as 234.56 and COBOL would display the value as \$34.56. Avoid this problem by defining map fields long enough to display all editing characters plus the largest possible value.

- On OS/400, the RSC option FILETYPE=SPOOL can be used on EZEPRINT file to inhibit the extra page eject.
- For VSE in CSP/AE 3.2.2 or CSP/AD 3.3, the default print destination was SYSLST. In VisualAge Generator Server for MVS, VSE, and VM, the print destination is the resource that is associated to the file name EZEPRINT in the resource association part. Thus, the default print destination for VSE batch is a VSE/POWER LST queue member with the jobname EZEPRINT. In VisualAge Generator Server for MVS, VSE, and VM, to set the print destination to SYSLST, you must specify the file name EZEPRINT to be file type SEQ in the resource association part and set up the runtime JCL. For more information on setting up your print destination, refer to the *VisualAge Generator Server Guide for MVS, VSE, and VM*.

CONVERSE Process Option

Migration considerations for the CONVERSE process option are as follows:

- Execution mode can be changed dynamically by the program, using the EZESEGM special function word. The execution mode of *either* is converted to *segmented* for main transactions and to *nonsegmented* for all other types.

The execution mode is converted when the program is imported into a VisualAge Generator library or migrated using the VisualAge Generator Migration Assistance Tool. If a program that is not a main transaction has an execution mode other than nonsegmented, it is changed to nonsegmented.

Under CSP/AD 3.2.2 or CSP/AD 3.3, execution mode could be overridden at generation either online or in batch using the generation option EXECMODE. In VisualAge Generator, this option is specified during program definition, instead of at generation time.

- In VisualAge Generator, the default value for the segmented transaction name in CICS is the name of the currently active transaction.
- Segmented execution mode is supported for MVS/TSO and VM CMS.

EZE Special Function Words

Migration considerations for EZE special function words are as follows:

- The value of the EZEORDER special function word is ignored unless the program is generated to perform numeric overflow checking using the /NUMOVFL generation option. If an overflow occurs, the value of the result is unpredictable.
- If the program logic is dependent on the specific values returned in the special function word EZERT8, specify the /NOSYSCODES generation option when generating existing programs.

If you specify /NOSYSCODES, the return codes are the Cross System Product codes returned by CSP/AE. The Cross System Product codes are consistent with the codes returned by applications running on releases of the Cross System Product set prior to version 4.1, but might not provide as much information as the system return codes.

- Special function word EZESEGM is supported in MVS/TSO and VM CMS.
- EZESEGTR is initialized to the CICS transaction name if a segmented transaction name is not explicitly specified at generation.
- EZEUSR in the MVS batch and VSE batch environments is set to the job name from the JOB card. EZEUSR in the VM batch environment is set to the user ID that is running the program.
- The requirement to always pass EZEDLPSB on a CALL for an IMS program has been removed. You must either pass the whole PSB, or pass the individual PCBs needed in the called program. For example:

```
CALL PGM1 EZEDLPSB;  
CALL PGM1 EZEDLPCB(0),EZEDLPCB(4),EZEDLPCB(5);
```
- EZECONVCM is the same as EZEDLTRM. Setting one of the language elements also sets the other. EZEDLTRM is available only for compatibility with releases of the Cross System Product set prior to version 3.3.

Transferring Among Programs

Migration considerations for transferring among programs are as follows:

- The NONCSP option on the XFER statement is used only for documentation and compatibility with the Cross System Product set.
- An XFER statement in a called program is not supported in any environment. Restructure programs in a run unit so that the XFER statement is performed by the main program.
- If a program is defined as segmented or single segment, a commit is taken on an XFER statement in the MVS/TSO or VM CMS environments. This is the same as in the IMS/VS and CICS environments.
- The NONCSP option must be specified for DXFR statements that transfer to non-VisualAge Generator programs, either as an option on the DXFR statement or in a DXFRLINK specification in the linkage table part.

Note: VSE batch does not support DXFR statements to non-VisualAge Generator programs.

- If program A calls program B, A must pass the same length of data that B expects. In CSP/AE, different lengths continue to run; in generated COBOL, errors are returned at runtime.
- Recursive program calls (for example: A calls B, which in turn calls A) are not supported. Recursive use of processes and statement groups in a program is supported.
- For CICS, the default segmented transaction name for program segments is the current CICS transaction name at the time of the CONVERSE. If the program issuing the CONVERSE process option was started via XCTL from a non-VisualAge Generator program, a separate transaction code should be defined for the program and specified as the segmented transaction name at program generation or moved to EZESEGTR prior to the CONVERSE.
- PL/I subroutines called by a program must have OPTION(COBOL) specified rather than OPTION(MAIN).
- COBOL 68 programs that call or are called by VisualAge Generator programs need to be migrated to COBOL for MVS and VM or to COBOL for VSE. For example, if the COBOL 68 program uses the SERVICE RELOAD instructions, it must be changed to use SET ADDRESS.

Programming Statements

The following statements have migration considerations:

- In CSP/AD 3.3 or earlier, the following statement was allowed in generation but not allowed in the test facility. In VisualAge Generator, it results in an error message.

```
IF record.item IS NULL.
```

File Support

Migration considerations for file and database support are as follows:

- If a record item is NULL, it was tolerated during generation for non-SQL records, but it failed in ITF for Version 3.3. It is no longer tolerated during generation.
- The file position (for serial, indexed, or relative files) after an unsuccessful INQUIRY, UPDATE, SCAN, or SCANBACK is undefined. The program must establish file position again when an unsuccessful read occurs.
- SCAN position is set on any successful I/O to the file. SCAN after any successful I/O operation retrieves the first record whose key is greater than the key of the record accessed on the previous I/O operation. If the record accessed on the previous I/O operation was the last record in the file, SCAN returns end-of-file (EOF).
- SET record SCAN results in an error message if used for serial or SQL records. Under CSP/AE, SET record SCAN was ignored.
- A SCANBACK process on a file that is not initialized returns NRF. A SCANBACK process on an empty file causes an EOF for non-CICS environments and both an EOF and NRF for CICS environments. A file is not initialized if it has never had any records in it. An empty file is one from which all records have been deleted.
- SCANBACK position is set on any successful I/O to the file. SCANBACK, after any successful I/O operation, retrieves the record with the highest key value that is less than the key of the record accessed on the previous I/O operation.
- The first SCANBACK does not have to be preceded by a SET record SCAN or any other I/O process. If SCANBACK is not preceded by another process, SCANBACK returns the last record in the file.
- SCANBACK following a SET record SCAN retrieves the record with the highest key value that is less than or equal to the current record key value. A SET record SCAN with a key value set to all X'FF' bytes prior to a SCANBACK sets the position to the end of the file. This causes the next SCANBACK to retrieve the last record in the file.
- SCANBACK following a SCAN that returned an EOF retrieves the last record in the file.
- SCAN following a SCANBACK retrieves the record following the record accessed on the SCANBACK. If SCANBACK returns an EOF, the SCAN returns the first record of the file.
- In CSP/AE, the outcome of a SCAN process option for a relative record varied depending on whether the application included other process options for the same record. If other options were included, a SCAN resulted in the current key being incremented by 1 and a direct read being issued for that key. NRF was returned if the record being read was deleted

or past the end of the file. However, EOF was never returned. The SCAN position could be changed by changing the relative record key without doing a SET record SCAN. If other options were not included, deleted records were skipped, current position was the record last read, and EOF was returned at the end of the file.

In VisualAge Generator, SCANS for relative records work consistently, regardless of what other process options are used with the record. SCANS for relative records do not return NRF if the next record is deleted. Instead, a SCAN always skips deleted records and retrieves the next record in the file based on the position set by the last successful I/O operation on the file. Both EOF and NRF are returned at the end of the file. The initial position is the beginning of the file. The SCAN position for relative records can only be changed by doing a successful I/O.

- When using alternate indexes, duplicate keys are handled as follows:
 - A SCAN process returns the record in the file with the next higher alternate key than the current position in the file. A DUP condition occurs if the record retrieved using SCAN has the same key as another record in the file. An exception occurs when retrieving the last record in a group of duplicate-keyed records. In this case, although the record has a duplicate key, the DUP mnemonic is not set.
 - If records with duplicate keys exist in the file, a SCAN following a SCAN retrieves any duplicate-keyed record before retrieving the record with the next key. Records with duplicate keys are returned in the order that VSAM returns them.
 - A SCAN following a successful process option other than SCAN skips over any duplicate-keyed records and retrieves the record with the next greater key.
 - A SCANBACK process returns the record in the file with the highest alternate key that is less than the current position in the file. A DUP condition occurs if the record retrieved using SCANBACK has the same key as another record in the file. An exception occurs when retrieving the last record in a group of duplicate-keyed records. In this case, although the record has a duplicate key, the DUP mnemonic is not set.
 - If records with duplicate keys exist in the file, a SCANBACK following a SCANBACK retrieves any duplicate-keyed record before retrieving the record with the previous key. Records with duplicate keys are returned in the order that VSAM returns them.
 - A SCANBACK following a successful process option other than SCANBACK skips over any duplicate-keyed records and retrieves the record with the next lower key.
- File names for files implemented as SEQ or VSAM in non-CICS environments cannot be COBOL reserved words or contain a \$, @, or # character. Use SEQRS on MVS and VM or VSAMRS on MVS, VSE, and VM for files that violate these restrictions to avoid changing the file name.

- If any program uses an alternate index to access a file generated with file type VSAM, then all the programs in the same job step that perform I/O to the same file must include the alternate specification record for the file, either as a process object or in the Tables and Additional Record list.
- In a run unit, only use the same file name for different records if all programs associate the file name with the same physical file.
- In a run unit, all records with the same file name must have the same attributes (record format, length, organization, key length, and key offset). They must also match the physical file definition.

Relational Database (SQL) Support

Migration considerations for SQL programs are as follows:

- Data items must be defined with a variable-length SQL code for variable-length data values to be written to the table. Data items must also specify the same length to VisualAge Generator as defined in the SQL table.
- The following SQL data codes are not supported for HEX data items:
 - 460 and 461, which are varying length, optionally null-terminated characters. The VisualAge Generator equivalent is CHA data with no SQL data code specified.
 - 484 and 485, decimal. The number of decimals cannot be specified for HEX data items. The VisualAge Generator equivalent is PACK data with no SQL data code specified.
- Large programs can result in a generated COBOL program that exceeds the SQL statement precompiler limits, depending on your SQL product.

The following are examples of precompiler limits:

- Maximum number of processed lines—All SQL statements must occur in the program prior to this limit. COBOL generation places the SQL statements as early as possible in the Procedure Division. However, programs might encounter this limit if they have many SQL processes or large numbers of data items in records or on maps.
- Maximum number of unique host variables—Each host variable that allows nulls also has an indicator variable that counts toward the maximum.
- Maximum number of lines/characters for an SQL statement.

Refer to the DB2 documentation for more information on precompiler limits for processed lines or unique host variables.

- All SQL statements are generated directly into the COBOL program. There is no dynamic SQL runtime option for programs. The only SQL statements that run dynamically at run time are those that have a dynamic table name

(table name in the SQL row record is a host variable) or those for which the **Execution Time Statement Build** option was specified in the SQL statement definition.

Any application that does not run in static mode with CSP/AE will not work as a generated COBOL program in VisualAge Generator without being modified to specify a dynamic table name or **Execution Time Statement Build** option for any statement that must be run dynamically. Using dynamic table names is the preferred option because **Execution Time Statement Build** has effects on the operation of other character host variables within the SQL statement.

- VisualAge Generator is more strict than Cross System Product in matching UPDATE or SETUPD processes with a REPLACE process. A problem occurs when there is more than one UPDATE or SETUPD for the same process object. Follow these steps to correct the problem:
 1. Edit the REPLACE process.
 2. Edit the SQL statement.
 3. Select the **Properties** tab and specify the name of the UPDATE or SETUPD process that corresponds to the REPLACE process in the **UPDATE/SETUPD process name** field.
- In some releases of Cross System Product, there was a problem that caused an incorrect SQL code to be stored. Follow these steps to correct the problem:
 1. Edit the SQL row record.
 2. From the **File** menu, select the **Save As** option and specify the existing record name to resave the record with the correct SQL code.

DL/I Database Support

Migration considerations for DL/I programs are as follows:

- The PSB name must be specified for the first program in an MVS/TSO, MVS batch, or VSE batch run unit if any program in the run unit needs to access DL/I. This was not required in CSP/AE.
- MVS/TSO and MVS batch applications with a PSB need to have at least two PCBs of any type in the PSB.
- If a process and a program that uses the process specify different PSB names, all the PCBs and segments in the process's PSB must be included in the program's PSB.

Data Values

Migration considerations for data values are as follows:

- The primary working storage record is initialized according to data type. In CSP/AE, it was initialized to blanks.

- Initialization of data items in serial, indexed, relative, DL/I segment, and SQL row records is based on the /INITRECD generation option. If /INITRECD is specified, the lowest-level data items (items with no substructure or columns in SQL rows) are initialized based on the data item type (blanks for CHA, DBCS, or MIX data, and zero for numeric data). If the /NOINITRECD generation option is specified, the records are not initialized. Initialization of the following is not controlled by the /INITRECD generation option:
 - Any record received as a parameter by a called program is not initialized.
 - Redefined records are not initialized.
 - The primary working storage is always initialized.
 - Other working storage records are initialized based on the data item type if /INITADDWS is specified as a generation option. If /NOINITADDWS is specified, the other working storage records are not initialized. /INITADDWS is the default.
- The /MATH generation option specifies whether the program should use the CSP/AE or the COBOL method of truncating intermediate results. The VisualAge Generator Developer test facility emulates the COBOL method. If the /MATH=COBOL generation option is specified, the results of arithmetic operations for generated COBOL programs might vary among the MVS, VSE, VM, and programmable workstation environments, depending on the COBOL compiler's implementation of arithmetic functions.

The value of the result of an arithmetic statement is unpredictable if overflow occurs. Maximum value overflow occurs whenever an intermediate result in an arithmetic calculation exceeds 18 significant digits. This means that overflow can occur more often for generated COBOL programs than for programs that are run using CSP/AE. If the /MATH=CSPAE generation option is specified and a calculation or an assignment statement is done, one of the following occurs:

 - The calculation in the generated COBOL program provides the same results as the application run using CSP/AE.
 - Overflow

To guarantee that results are the same, specify the /MATH=CSPAE and /NUMOVFL generation options, and use EZEORDER and EZEORDER to check for overflow. Alternatively, specify /MATH=CSPAE and define all the variables so that overflow cannot occur.
- Operations on items not initialized or on items with data that is not valid can have different results (including ABENDS) in generated COBOL programs than they have in CSP/AE. This includes operations on numeric items initialized to blanks. The /SPZERO option can be used to treat blanks

as zeros in generated-COBOL applications. However, using the /SPZERO option to treat blanks as zeros makes the program bigger and slower. /SPZERO only supports NUM and NUMC data items. It does not support PACK or PACF data items.

User Message Files

Migration considerations for user message files are as follows:

- In CSP/AE, user messages were stored in files or databases outside of the product. In VisualAge Generator they are stored in message tables and generated into COBOL programs. Existing files must be converted. The message file conversion utility can be used to convert message files to message tables. See “Appendix F. Using the Message File Conversion Utility” on page 123 and the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information on the message file conversion utility.
Message tables eliminate the need for maintaining special message files and databases and allow messages to be displayed with or without the message identifier.
- The message file identifier in the program specification must be at least 3 characters long.
- Under CSP/AD and CSP/AE, a four-character message file ID in the application specification resulted in the fourth character being moved into the message prefix at run time to assist in identifying the application. This does not occur in VisualAge Generator. As a result, each program must do one of the following:
 - Have its own message table
 - Drop the use of the fourth character, if you want to use a single table
- When you print a program, only the item list for the message table is printed; the table contents are not printed. To print the table contents, print the message table by itself.

Language-Dependent Variables

Under CSP/AD 3.3 and CSP/AE 3.3, message 1 was used to indicate the language-dependent variables (YES/NO response, currency symbol, numeric separator, and SQL indicators). With VisualAge Generator Server for MVS, VSE, and VM, the language-dependent variables are in a language-dependent installation module. You can customize the module when you install VisualAge Generator Server for MVS, VSE, and VM. For additional information about setting the language defaults for your installation, refer to the *Program Directory for VisualAge Generator Server* for your MVS, VSE or VM environment and the *VisualAge Generator Server Guide for MVS, VSE, and VM*.

CSP/AD Language Elements Associated with the Application Load File

CSP/AD language elements associated with the CSP/AE load modules in the ALF are not supported.

Application Storage Requirements

The application storage requirements function of CSP/AD 3.3 and earlier releases reported the size of CSP/AE application load modules in the ALF. This function is not required with VisualAge Generator Server for MVS, VSE, and VM because the load modules are standard COBOL.

Because a program has become a standard COBOL load module, when a program is running, it now is loaded into memory instead of being read into GETMAIN storage. This means that the residency of the program is controlled by the target environment.

In ESA systems for MVS CICS and VSE CICS, by default, the load modules are loaded above the 16MB line. In MVS CICS and VSE CICS, the load modules remain in storage until the region is shut down, regardless of whether the modules are marked as resident, because these releases of CICS do not perform program compression above the 16MB line. Depending on the size of your programs, the number of programs you have, and the size of your region, you could receive an insufficient storage condition. Therefore, you should ensure that your “above the line” region is large enough to contain the load modules used over the life of a region, or link all the programs to run below the 16MB line.

In CICS/ESA, this problem does not exist.

In VM/ESA, the storage needed by the program plus the storage needed for VisualAge Generator Server for MVS, VSE, and VM determines the virtual machine size needed to run a VisualAge Generator 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.

A virtual machine size of 20MB (megabytes) is recommended for running programs using VisualAge Generator Server for MVS, VSE, and VM, and is necessary if you want the VisualAge Generator Server for MVS, VSE, and VM code to run above the 16MB line. Running VisualAge Generator Server for MVS, VSE, and VM above the 16MB line allows more room for VisualAge Generator programs below the 16MB line.

Coexistence with CSP/AE Applications

The applications, map groups, and tables managed by CSP/AE are not the same objects created by generating VisualAge Generator programs, map groups, and tables.

CSP/AE applications and generated COBOL programs can coexist on the same system. However, generated COBOL programs look like non-Cross System Product programs to CSP/AE. Therefore, to avoid inconsistent operating results, it is better to convert all applications that run together in the same transaction, job step, or MVS/TSO or VM invocation from CSP/AE run time to COBOL run time at the same time. Operating inconsistencies occur on partial conversions from CSP/AE to COBOL run time in sets of interacting CSP/AE applications and generated COBOL programs with the following characteristics:

Note: In the following list, the word "programs" refers to both CSP/AE applications and generated COBOL programs.

- The programs access the same file or database.
- Called programs depend on position being maintained in a file or database across multiple calls to the program.
- The programs access the same VisualAge Generator tables.
- Multiple programs do printing or the single report is printed using multiple calls to the same program.
- Any of the programs does an explicit CALL to the EZECOMIT or EZEROLLB service routines.
- Multiple programs do CONVERSE processing.

If the above characteristics do not apply in your situation, generated COBOL programs can interface with CSP/AE applications in the same way the non-Cross System Product programs interface with CSP/AE applications.

CSP/AE can call a generated COBOL program if that called program has been generated with default linkage and the called program is not found in the CSP/AE ALF.

A generated COBOL program can call a CSP/AE application only through a bridge program. The bridge program expects a generated COBOL linkage and calls CSP/AE as defined in the CSP/AE 3.3 (or earlier) documentation for calls from a non-Cross System Product program to a CSP/AE application. There is no COBOL linkage table entry that directly generates a call to CSP/AE applications.

A CSP/AE application can coexist on the same system with the corresponding VisualAge Generator program generated as a COBOL program. This allows the same source code to be generated for use with both CSP/AD applications and generated COBOL programs. If you need to do this, make all changes to the called application source in CSP/AD. Then export the external source format from CSP/AD, import it into VisualAge Generator, and generate the COBOL program. After you have migrated everything to VisualAge Generator, you can stop maintaining the CSP/AD source and only make updates to the VisualAge Generator source.

If CSP/AE applications use a table or map group that is also generated as a VisualAge Generator table or map group, the CSP/AE applications do not share the table or map group with generated COBOL programs.

For OS/400 coexistence restrictions, refer to the *Running VisualGen Applications on OS/400* document.

Calls or Transfers from Non-CSP Programs

Non-Cross System Product programs that currently call or transfer to CSP/AE applications must be modified if the CSP/AE applications are generated again as COBOL. The modifications include calling or transferring directly to the generated COBOL program instead of starting up the CSP/AE interpreter and passing the name of the application to be run as a parameter. Refer to the *VisualAge Generator Client/Server Communications Guide* for a detailed description of the interface to programs generated with the VisualAge Generator generation facility.

Transferring Control in the CICS Environment

For the CICS environment, the X'FFFFFFFF' fullword at the end of the parameter list is consistent with the CSP/AE CALL interface. For generated COBOL programs, the length of the COMMAREA does not include this fullword unless /ENDCOMMAREA is specified as a generation option for the calling program. If the generation option /ENDCOMMAREA was specified and the parameter format COMMPTR is in effect, the length specified for COMMAREA on the EXEC CICS LINK command is automatically increased by 4 bytes. Under certain conditions, CICS passes a copy of the COMMAREA to the called program. Specify /ENDCOMMAREA to ensure that the X'FFFFFFFF' fullword is included when a copy of the COMMAREA is made.

MVS CICS and VSE CICS Resource Table Requirements for Programs

To install generated programs for MVS CICS and VSE CICS, all main transaction, main batch, dynamically called programs, print services programs, map group format modules, and table programs must have an entry in the PPT (or the RDO equivalent). To start each main transaction program by entering a transaction code started from MVS CICS and VSE CICS, or with XFER, CREATX, or EZESEGTR as the result of a segmented converse, each program must have a unique transaction ID assigned and have an entry in the PCT (or the RDO equivalent). Specify /CICSENTRIES=RDO as a generation option if you want to generate model resource definition online (RDO) program and transaction definitions. Specify /CICSENTRIES=MACRO as a generation option if you want to generate model PPT and PCT table entries.

Refer to the *VisualAge Generator Server Guide for MVS, VSE, and VM* for additional information on installing generated programs.

MVS CICS and VSE CICS Table Entries for VisualAge Generator Server for MVS, VSE, and VM

If you previously installed CSP/AE and plan to run CSP/AE and VisualAge Generator Server for MVS, VSE, and VM in the same MVS CICS or VSE CICS region, you need to make the following modifications to your MVS CICS or VSE CICS tables and JCL:

- Remove the definitions for the extrapartition queue from the CSP/AE DCT source with DESTID=ETLP.
- Remove the definitions for the indirect queues from the CSP/AE DCT with DESTID=EZEP and DESTID=EZET.
- Remove the definitions for the SDSCI entry from the indirect queues with DSCNAME=ASAPR.
- Change the TRANSID parameters for any DCT INTRA queues with TRANSID=XSPZ to TRANSID=EZEP and TRANSID=XSPZ to TRANSID=EZEZ. The VisualAge Generator Server for MVS, VSE, and VM transactions EZEP and EZEZ are compatible with the CSP/AE XSPZ and XSPZ transactions and should be used instead of XSPZ and XSPZ.
- Remove the ASAPR DD statement from your MVS CICS or VSE CICS startup JCLs. Ensure the appropriate MVS CICS or VSE CICS tables have been updated to add support for COBOL. For a complete list of the MVS CICS or VSE CICS table entries for VisualAge Generator Server for MVS, VSE, and VM, refer to the *Program Directory for VisualAge Generator Server* for your MVS, VSE or VM environment.

National Language Codes

The NLS code has been extended to three characters. The first two characters define the base language (for example, EN is for English). The third character can either represent natural continuation or identifies variation, script, or dialect. The NLS codes are as follows:

Table 8. Language Support Suffixes

Description	CSP V3.3	VisualAge Generator
Default language	D	See note.
U.S. English (Mixed case English)	E	ENU
Uppercase English	U	ENP
Simplified Chinese	C	CHS
German (Germany)	G	DEU
German (Switzerland)	W	DES
Spanish	n/a	ESP
Japanese (Katakana)	J	JPN
Korean	K	KOR
Portuguese (Brazilian)	P	PTB

Note: The default language is specified as an installation option. A unique code for the default language is no longer required.

CSP/AE Invocation Parameters

CSP/AE invocation parameters have been replaced with generation options, installation options, or are no longer supported. The invocation parameter group file is still used by the FZETPRT utility program.

MVS CICS Invocation Parameters

Table 9 shows how MVS CICS invocation parameters are supported when using generated COBOL programs.

Table 9. Invocation Parameter Migration for MVS CICS

Parameter	Corresponding Function in VisualAge Generator
P=yyyy	You can specify an alternate transient data or spool file during generation by using a resource association part. The program can dynamically change the target transient data queue name or JES spool file name by using the special function word EZEDESTP.

Table 9. Invocation Parameter Migration for MVS CICS (continued)

Parameter	Corresponding Function in VisualAge Generator
NLS=n	The NLS code for VisualAge Generator Server for MVS, VSE, and VM is specified at generation time with the /TARGNLS generation option.
SEG	The application switches between segmented and nonsegmented mode by setting EZESEGM.
DMODE=S D A	You choose whether DB2 or ANSI SQL statements are generated using the /ANSISQL generation option. DB2 SQL statements or ANSI SQL statements are imbedded directly into the generated COBOL program. DMODE=D (dynamic execution without a program-specific BIND) is not supported.
RT=zzzz	The RT option is specified at generation time through the /RT generation option. The user no longer has to press attention interrupt to go on to the next transaction; the transaction is started automatically.
TSMS	The /WORKDB generation option specifies whether auxiliary or main temporary storage is used for saving working storage across segments.
NOTXA	Use the /DATA=24 generation option to force all working storage to be acquired below the line.
IMSESA	The IMSESA setting is controlled by the installation options module.
FFFF	The /ENDCOMMAREA generation option specifies whether the end of list indicator is used.
PARMS=xxxxxxx	Invocation Parameter Group name is not supported.

MVS/TSO and MVS Batch Invocation Parameters

Table 10 shows how MVS/TSO invocation parameters are supported when using generated COBOL programs.

Table 10. Invocation Parameter Migration for MVS/TSO and MVS Batch

Parameter	Corresponding Function in VisualAge Generator
NLS(n)	The NLS code for VisualAge Generator Server for MVS, VSE, and VM is specified at generation time with the /TARGNLS generation option.
DSYS(ssss)	The DB2 subsystem is specified in the CLIST. You can modify the generated CLIST to change the DB2 subsystem used with the program. The system administrator could also modify the CLIST template to set the default DB2 subsystem ID to the subsystem that is normally used.

Table 10. Invocation Parameter Migration for MVS/TSO and MVS Batch (continued)

Parameter	Corresponding Function in VisualAge Generator
PSB(pppppppp)	The PSB name is specified in the CLIST. The name defaults to the name of the program's PSB. You can modify the generated CLIST to change the PSB value specified for the program. The system administrator could also modify the CLIST template to set the default PSB name generated in the CLIST to be the program name instead of the program's PSB name.
BKO(xxx)	The BKO value is specified in the CLIST. You can modify the generated CLIST to change the BKO value specified for the program. The system administrator could also modify the CLIST template to change the BKO value specified for the program.
DMODE=S D A	You choose whether DB2 or ANSI SQL statements are generated using the /ANSISQL generation option. DB2 SQL statements or ANSI SQL statements are imbedded directly into the generated COBOL program. DMODE=D (dynamic execution without a program-specific BIND) is not supported.
DPLAN(pppppppp)	The DB2 plan name is specified in the CLIST. The default plan name is the program name. You can modify the generated CLIST to change the DB2 plan used with the program.
PARMS=xxxxxxxx	Invocation Parameter Group name is not supported.

For MVS batch applications, the NLS, DMODE, and PARMS invocation parameters are handled as shown in Table 10 on page 99 for MVS/TSO. The U invocation parameter for MVS batch applications is not supported. The EZEUSR and EZEUSRID statements are always set to the job name from the job card.

In the MVS/TSO environment, a generated main transaction or main batch program is invoked by a CLIST. A sample CLIST specifically for the program is created by the COBOL generator if the /RUNFILE option is specified. The CLIST might need to be modified to add data set ALLOCATE commands for data sets required by called or transferred-to programs or for data sets accessed by setting the EZEDEST and EZEDESTP special function words.

VSE CICS and VSE Batch Invocation Parameters

Table 11 on page 101 shows how VSE CICS invocation parameters are supported when using generated COBOL applications.

Table 11. Invocation Parameter Migration for VSE CICS and VSE Batch

Parameter	Corresponding Function in VisualAge Generator
P=yyyy	You can specify an alternate transient data or spool file during generation by using a resource association part. The program can dynamically change the target transient data queue name or POWER spool file name by using the special function word EZEDESTP.
NLS=n	The NLS code for VisualAge Generator Server for MVS, VSE, and VM is specified at generation time with the /TARGNLS generation option.
SEG	The application switches between segmented and nonsegmented mode by setting EZESEGM.
SMODE=S D A	You choose whether DB2/VSE or ANSI SQL statements are generated using the /ANSISQL generation option. DB2/VSE SQL statements or ANSI SQL statements are imbedded directly into the generated COBOL program. SMODE=D (dynamic execution) is not supported.
SID	<p>All of the SQL calls are handled by COBOL statements. For VSE CICS, DB2/VSE automatically connects the user to the database at the first SQL call using the implicit user ID for CICS. If the user is signed on as a VSE/ESA interactive user interface user, the implicit CICS user ID is connected with that VSE/ESA user ID. Otherwise, it is the CICS default set by the CIRB transaction. If you want to use an ID other than the implicit ID, your program must issue an EZECONCT call with the desired user ID and password prior to the first SQL process option.</p> <p>For VSE batch, VisualAge Generator Server for MVS, VSE, and VM has to connect the user to the database before the first SQL call. The SID=userid/password parameter specified in SYSIPT of the job is used to perform this database connection.</p>
RT=zzzz	The RT option is specified at generation time through the /RT generation option. The user no longer has to press attention interrupt to go on to the next transaction; the transaction is started automatically.
TSMS	The /WORKDB generation option specifies whether auxiliary or main temporary storage is used for saving working storage across segments.
FFFF	The /ENDCOMMAREA generation option specifies whether the end of list indicator is used.

For VSE batch applications, the NLS, SMODE, SID, and PARMs invocation parameters are handled as shown in Table 11 for VSE CICS. The U invocation parameter for VSE batch applications is not supported. The EZEUSR and EZEUSRID statements are always set to the job name from the job card.

VM Invocation Parameters

Table 12 shows how VM invocation parameters are supported when using generated COBOL applications.

Table 12. Invocation Parameter Migration for VM

Parameter	Corresponding Function in VisualAge Generator
P=filename	You can specify a serial or print file during generation by using a resource association part. The program can dynamically change the target serial file or print file by using the special function word EZEDESTP.
NLS=X	The NLS code for VisualAge Generator Server for MVS, VSE, and VM is specified at generation time with the /TARGNLS generation option.
LL=xxxxxxx	There is no equivalent of the CSP LL= invocation parameter. A CALL statement to the non-VisualAge Generator program is generated in the COBOL program. This causes the CMS search order to be searched for a CMS module that matches the non-VisualAge Generator program name. If one is not found, the LOADLIBs in the GLOBAL LOADLIB list is searched for a match. If still no match is found, an attempt is made to find a TEXT file with same file name.
SMODE=M S D A DM DS MD SD	For SMODE=A, you can choose whether SQL/DS or ANSI SQL statements are generated using the /ANSISQL generation option. SQL/DS statements or ANSI SQL statements are imbedded directly into the generated COBOL program. SMODE=D (dynamic execution) is not supported. SMODE=S (single-user mode) or SMODE=M (multi-user mode) are defined by the user-defined symbolic parameter SQLSTMDE and are used during program generation. The default is SQLSTMDE=MULTIUSER. If SQLSTMDE=SINGLEUSER, the user-defined symbolic parameter SQLSTOPT can be used to specify the file name of a file containing SQL/DS startup options. Other SQL/DS user-defined symbolic parameters are SQLDBNAM, SQLPKGNM, SQLPROPT, and SQLUSRPW. Refer to the <i>VisualAge Generator Generation Guide</i> and the <i>VisualAge Generator Server Guide for MVS, VSE, and VM</i> for more information.

Table 12. Invocation Parameter Migration for VM (continued)

Parameter	Corresponding Function in VisualAge Generator
SID=	<p>All of the SQL calls are handled by COBOL statements. In VM, there are several ways to handle the connection to the SQL database.</p> <p>For VM, if no user ID and password are specified, SQL/DS connects the user to the database at the first SQL call using the VM user ID and password.</p> <p>If you want to use an ID other than the implicit ID, your program can issue an EZECONCT call with the desired user ID and password prior to the first SQL process option.</p> <p>During generation, you can use the SQLUSRPW symbolic parameter to specify the user ID and password to be used for the SQL connection.</p> <p>In addition, the user can specify the SID= parameter on the program runtime REXX exec to specify the user ID and password to be used for the database connection.</p> <p>Refer to the <i>VisualAge Generator Generation Guide</i> and the <i>VisualAge Generator Server Guide for MVS, VSE, and VM</i> for more information.</p>
SDB=	<p>For VM, if no database name is specified, SQL/DS connects the user to the database at the first SQL call using the database specified on the last SQLINIT issued for the virtual machine.</p> <p>If you want to use a database other than the implicit database, your program can issue an EZECONCT call with the desired user ID, password, and database name prior to the first SQL process option.</p> <p>During generation, you can use the SQLDBNAM symbolic parameter to specify the database name to be used for the SQL connection.</p> <p>In addition, the user can specify the SDB= parameter on the program runtime REXX exec to specify the database name to be used for the database connection.</p> <p>Refer to the <i>VisualAge Generator Generation Guide</i> and the <i>VisualAge Generator Server Guide for MVS, VSE, and VM</i> for more information.</p>
ALF=C V	<p>VisualAge Generator Server for MVS, VSE, and VM does not have an equivalent of the ALF parameter. All programs reside in the CMS LOADLIB. The LOADLIB name is specified in the runtime exec.</p>

Table 12. Invocation Parameter Migration for VM (continued)

Parameter	Corresponding Function in VisualAge Generator
NOTXA	Use the /DATA=24 generation option to force all working storage to be acquired below the line.
SS=YES NO	VisualAge Generator Server for MVS, VSE, and VM product code can be loaded from the product LOADLIB or from a saved shared segment. You make the decision during VisualAge Generator installation.

In the VM CMS and VM batch environments, a generated main transaction or main batch program is invoked by a runtime REXX exec. A sample runtime exec specifically for the program is created by the COBOL generator if the /RUNFILE option is specified. The sample runtime REXX exec might need to be modified before it is used. Refer to the *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information on tailoring the sample runtime REXX exec.

National Language Applications

When installing different national language versions of an application on the same CICS system using CSP/AE, you generate each version into a different ALF and have the user select the appropriate ALF when starting the transaction. In VisualAge Generator, you must change the program name and assign a separate transaction code for each national language version of the program.

Map groups and language-dependent tables also need to have different names for each language. This is because programs in MVS CICS, VSE CICS, and CICS OS/2 systems must have unique names.

Non-VisualAge Generator Interface to DCBRINIT on MVS CICS and VSE CICS CONVERSEs

If you are migrating to VisualAge Generator from CSP/AE, you might have used a non-Cross System Product program to interface to either DCBINIT (the CSP/AE interpreter) or to DCBRINIT (the CSP/AE routine to restart after a segmented CONVERSE). In CSP/370AD 4.1 and CSP/370RS 2.1, if you used both CSP/AE applications and generated COBOL programs or you used only generated COBOL programs and you wanted to continue to run a non-VisualAge Generator program during a segment break, the module ELATSRST was used in place of DCBRINIT. VisualAge Generator also provides the ELATSRST module.

Some CSP/AE customers developed systems whereby the initial program receiving control from CICS was always a non-CSP/AE program. This occurred even when the transaction was a segment started following the CONVERSE process option from a pseudoconversational application. The non-CSP/AE program transfers control using the XCTL statement to DCBINIT (the CSP/AE interpreter) or DCBRINIT (CSP/AE interpreter segment restart) to continue processing using CSP/AE.

The techniques described in this section enable you to implement these functions using a VisualAge Generator Server for MVS, VSE, and VM module, ELATSRST, instead of DCBINIT or DCBRINIT. ELATSRST is provided as a migration aid. It should not be used for new programs. Consider the following before you use ELATSRST:

- This technique does not support the following:
 - ELATSRST during a transfer using an XFER statement with a Map or First Map processing
 - Programs targeted for the CICS OS/2 environment
 - Programs targeted for non-CICS environments on the host.
- There is performance overhead when using ELATSRST because:
 - There are two extra XCTLs
 - There is an extra read of the temporary storage queue where information is saved during the CONVERSE.

See Figure 9 on page 106 for an example of a CSP/AE application:

```

trancode X ----- tranocode Y
  PROGA      started by CICS
    - does some common, front-end processing
    - determines which application is associated with
      the transaction code
    - creates a COMMAREA with the ALF name, application name and
      possibly working storage
    - issues a CICS XCTL command to DCBINIT

  DCBINIT
    - starts the requested application
    - the application
      - sets EZSEGTR to a new transaction code (tranocode Y)
      - does a segmented CONVERSE

application user enters data
trancode Y -----

  PROGB      started by CICS
    - does some common, front-end processing
    - issues a CICS XCTL command to DCBRINIT

  DCBRINIT
    - determines which application to resume
    - resumes processing for the application after the
      segmented CONVERSE

```

Figure 9. Sample CSP/AE Application

See Figure 10 on page 107 for a sample technique with VisualAge Generator Server for MVS, VSE, and VM:

```

trancode X ----- tranocode Y
  PROGA      started by CICS
    - does some common, front-end processing
    - determines which application is associated with
      the transaction code
    - determines whether the application is generated for
      CSP/AE or as a COBOL program
    - if the application is generated for CSP/AE:
      - creates a COMMAREA with the ALF name, application name and
        possibly working storage
      - issues a CICS XCTL command to DCBINIT
    - if the program is generated as a COBOL program:
      - creates a COMMAREA containing the working storage record,
        if any, for the program
      - issues a CICS XCTL command to the program

  DCBINIT (if CSP/AE is being used)
    - starts the requested application
    - the application
      - sets EZESEGTR to a new transaction code (trancode Y)
      - does a segmented CONVERSE
  The COBOL program (if generated COBOL is being used):
    - sets EZESEGTR to a new transaction code (trancode Y)
    - does a segmented CONVERSE

application user enters data
trancode Y -----

  PROGB      started by CICS
    - does some common, front-end processing
    - issues a CICS XCTL command to ELATSRST

  ELATSRST
    - determines whether the restart is for an application
      generated for CSP/AE or a generated COBOL program
    - if it was for an application generated for CSP/AE
      - issues a CICS XCTL command to DCBRINIT
    - if it was for a generated COBOL program
      - issues a CICS XCTL command to the program
    - a CICS abend occurs if either the XCTL is unsuccessful or
      other CICS detectable conditions occur (such as AE10)
    - a VisualAge Generator Server for MVS, VSE, and VM abend, ELAF,
      occurs if VisualAge Generator Server detectable errors
      are found (for example, not resuming after a CONVERSE).
      Please see the VisualAge Generator Server Guide for MVS, VSE, and VM
      if this abend code occurs.

  DCBRINIT
    - determines which application to resume
    - resumes processing for the application after the
      segmented CONVERSE
  The COBOL program (if generated COBOL is being used):
    - resumes processing

```

Figure 10. VisualAge Generator Example Using ELATSRST

To implement the transfer using ELATSRST the following changes are needed:

- PROGA

This is the non-VisualAge Generator program that is started by entering a transaction code at a terminal or by issuing the CICS START command. The following changes are required:

- Have some mechanism that enables you to determine whether the transaction code is for an application generated for CSP/AE or a generated COBOL program.
- XCTL to DCBINIT if the application is generated for CSP/AE, using a COMMAREA in the following format (same as under CSP/AE):

```
EXEC CICS XCTL PROGRAM('DCBINIT') COMMAREA(COMWORK)
                                  LENGTH(length of COMWORK)
```

Data in the COMMAREA is as follows:

```
'useralf.applnam wsrdata.....'
```

where useralf is optional, 1-7 chars, must be followed by period if it is used
applnam is required and must be padded by blanks to 8 chars.
The application must be defined as a main application.
wsrdata is optional and must follow the 8-char applnam;
it is the data used to initialize the primary working storage record for the application

- If a generated COBOL program is being used and you want to initialize the program's primary working storage record, XCTL to the program using a COMMAREA in the following format:

```
EXEC CICS XCTL PROGRAM('pgmname') COMMAREA(COMWORK)
                                  LENGTH(length of COMWORK)
```

Data in the COMMAREA is as follows:

```
'wsrdata.....'
```

where pgmname is the name of the program

wsrdata is optional and is the data used to initialize the primary working storage record for the program

If you do not need to initialize the primary working storage record for the program, omit the COMMAREA and LENGTH parameters.

Note: If your non-VisualAge Generator program was started using an EXEC CICS START command, you are responsible for retrieving any data that was passed by the previous application or program and passing it through the COMMAREA if it is needed by the next application or program.

- PROGB

This is the non-VisualAge Generator program that is used when restarting after a segmented CONVERSE.

The changes that are required in PROGB are as follows:

- XCTL to ELATSRST, insuring that the COMMAREA contents and length that were received, are passed unchanged to ELATSRST. An example of the appropriate command is:

```
EXEC CICS XCTL PROGRAM('ELATSRST') COMMAREA(COMMAREA)
                                LENGTH(EIBCALEN)
```

Appendix C. Cross System Product Interpretive to C++

The following considerations apply when migrating from Cross System Product to VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX.

All C++ environments in VisualAge Generator (OS/2, AIX, CICS/6000, CICS for Windows NT, HP-UX, and Windows NT) are new. Native Oracle is supported in the OS/2, AIX, Windows NT, and HP-UX environments. For more information on compatibility considerations, refer to the *VisualAge Generator Programmer's Reference*.

Part Names

The following are considerations for part names:

- Program, map group, and table names:
 - Must be unique
 - Must not be C++ reserved words

Language Element Compatibility Considerations

The following gives an overview of the compatibility considerations for language elements. The language elements include the following:

- Mapping support
- CONVERSE process option
- EZE special function words
- Transferring among programs
- File support
- Relational database (SQL) support
- DL/I database support
- Data values
- User message files

Mapping Support

Migration considerations for mapping support are as follows:

- An automatic form feed for each printer file is issued whenever a main program ends or when a program called by a non-VisualAge Generator program returns to the calling program. An automatic form feed is also

issued for the current printer file when a CLOSE process option is explicitly issued. No form feed occurs when a program starts or when a called program returns to another VisualAge Generator program. The automatic form feed can be suppressed by adding an entry in the resource association part for EZEPRINT and specifying the /NOFF option.

- Only one definition of a floating area is supported for printers or terminal maps. If different floating areas are defined for different print devices (for example, PRINT, PRINT-B, 3767), the definition that is used depends on the order in which the maps were defined and cannot be predicted. To avoid confusion, either specify the same device for all print and terminal maps or specify the same floating area for all printers and terminal devices.
- The display of a negative zero value is different for CSP/AE and C++ applications. If a numeric variable field is defined with zero edit and a leading or trailing sign, and a value -0 is moved to the item, CSP/AE displays the negative sign. However, the same map field in the C++ program will not display the negative sign. If the field is defined with a trailing sign, it will be displayed as 0+ when running the C++ program. If the field is defined with a leading sign, it will be displayed as 0.

CONVERSE Process Option

Migration considerations for the CONVERSE process option are as follows:

- Execution mode can be changed dynamically by the program, using the EZESEGM special function word. The execution mode of *either* is converted to *segmented* for main transactions and to *nonsegmented* for all other types. The execution mode is converted when the program is imported into a VisualAge Generator library or migrated using the VisualAge Generator Migration Assistance Tool. If a program that is not a main transaction has an execution mode other than nonsegmented, it is changed to nonsegmented.

Under CSP/AD 3.2.2 or CSP/AD 3.3, execution mode could be overridden at generation either online or in batch using the generation option EXECMODE. In VisualAge Generator, this option is specified during program definition, instead of at generation time.

- In VisualAge Generator, the default value for the segmented transaction name in CICS is the name of the currently active transaction.

EZE Special Function Words

Migration considerations for EZE special function words are as follows:

- In VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX, the print destination is the resource that is associated to the file name EZEPRINT in the resource association part.
- The special function word EZERT8 always contains system dependent codes. The /NOSYSCODES generation option is not supported.

Transferring Among Programs

Migration considerations for transferring among programs are as follows:

- An XFER statement in a called program is not supported in any environment. Restructure programs in a run unit so that the XFER statement is performed by the main program.
- If a program is defined as segmented or single segment, a commit is taken on an XFER statement.
- The NONCSP generation option must be specified for DXFR statements that transfer to non-VisualAge Generator programs, either as an option on the DXFR statement or in a DXFRLINK specification in the linkage table part.
- Recursive program calls (for example: A calls B, which in turn calls A) are not supported. Recursive use of processes and statement groups in a program is supported.

File Support

Migration considerations for file and database support are as follows:

- The file position (for serial, indexed, or relative files) after an unsuccessful INQUIRY, UPDATE, SCAN, or SCANBACK is undefined. The program must establish file position again when an unsuccessful read occurs.
- SCAN position is set on any successful I/O to the file. SCAN after any successful I/O operation retrieves the first record whose key is greater than the key of the record accessed on the previous I/O operation. If the record accessed on the previous I/O operation was the last record in the file, SCAN returns end-of-file (EOF).
- SET record SCAN results in an error message if used for serial or SQL records. Under CSP/AE, SET record SCAN was ignored.
- A SCANBACK process on a file that is not initialized returns NRF. A SCANBACK process on an empty file causes an EOF for non-CICS environments and both an EOF and NRF for CICS environments. A file is not initialized if it has never had any records in it. An empty file is one from which all records have been deleted. In the native environments (non-CICS), a SCAN or SCANBACK process on a file that does not exist returns FNF.
- SCANBACK position is set on any successful I/O to the file. SCANBACK, after any successful I/O operation, retrieves the record with the highest key value that is less than the key of the record accessed on the previous I/O operation.
- SCANBACK following a SET record SCAN retrieves the record with the highest key value that is less than or equal to the current record key value. A SET record SCAN with a key value set to all X'FF' bytes prior to a SCANBACK sets the position to the end of the file. This causes the next SCANBACK to retrieve the last record in the file.

- SCANBACK following a SCAN that returned an EOF retrieves the last record in the file.
- SCAN following a SCANBACK retrieves the record following the record accessed on the SCANBACK.
- When using alternate indexes, duplicate keys are handled as follows:
 - A SCAN process returns the record in the file with the next higher alternate key than the current position in the file. A DUP condition occurs if the record retrieved using SCAN has the same key as another record in the file. An exception occurs when retrieving the last record in a group of duplicate-keyed records. In this case, although the record has a duplicate key, the DUP mnemonic is not set.
 - If records with duplicate keys exist in the file, a SCAN following a SCANBACK retrieves any duplicate-keyed record before retrieving the record with the next key. Records with duplicate keys are returned in the order that the file system returns them.
 - A SCANBACK process returns the record in the file with the highest alternate key that is less than the current position in the file. A DUP condition occurs if the record retrieved using SCANBACK has the same key as another record in the file. An exception occurs when retrieving the last record in a group of duplicate-keyed records. In this case, although the record has a duplicate key, the DUP mnemonic is not set.
 - If records with duplicate keys exist in the file, a SCANBACK following a SCANBACK retrieves any duplicate-keyed record before retrieving the record with the previous key. Records with duplicate keys are returned in the order that the file system returns them.

Relational Database (SQL) Support

Migration considerations for SQL programs are as follows:

- Data items must be defined with a variable-length SQL code for variable-length data values to be written to the table. Data items must also specify the same length to VisualAge Generator as defined in the SQL table.
- The following SQL data codes are not supported for HEX data items:
 - 460 and 461, which are varying length, optionally null-terminated characters. VisualAge Generator equivalent is CHA data with no SQL data code specified.
 - 484 and 485, decimal. The number of decimals cannot be specified for HEX data items. VisualAge Generator equivalent is PACK data with no SQL data code specified.
- Large programs can result in a generated C++ program that exceeds the DB2 SQL statement precompiler limits, depending on your release of DB2. The following are examples of precompiler limits:

- Maximum number of processed lines—All SQL statements must occur in the program prior to this limit.
- Maximum number of unique host variables—Each host variable that allows nulls also has an indicator variable that counts toward the maximum.
- Maximum number of lines/characters for an SQL statement

Refer to the DB2 documentation for more information on precompiler limits for processed lines or unique host variables.

- All SQL statements are generated directly into the C++ program. There is no dynamic SQL runtime option for programs. The only SQL statements that run dynamically at run time are those that have a dynamic table name (table name in the SQL row record is a host variable) or those for which the **Execution Time Statement Build** option was specified in the SQL statement definition.

Any application that does not run in static mode with CSP/AE will not work as a generated C++ program in VisualAge Generator without being modified to specify a dynamic table name or **Execution Time Statement Build** option for any statement that must be run dynamically. Using dynamic table names is the preferred option because **Execution Time Statement Build** has effects on the operation of other character host variables within the SQL statement.

- Several differences exist between the SQL support provided by the DB2 family of products on various platforms. These differences are listed in the *Formal Register of Extensions and Differences in SQL (SC26-3316)*.

DL/I Database Support

Migration considerations for DL/I programs are as follows:

- DL/I programs are not supported in the C++ environments.

Data Values

Migration considerations for data values are as follows:

- All records are initialized according to data type; in CSP/AE, they were initialized to blanks.
- If the program is a client/server program and runs on AIX, the **CONTABLE** parameter is required in the client linkage table in order to eliminate data alignment problems.

User Message Files

Migration considerations for user message files are as follows:

- In CSP/AE, user messages were stored in files or databases outside of the product. In VisualAge Generator they are stored in message tables and

generated into table files. Existing files must be converted. The message file conversion utility can be used to convert message files to message tables. See “Appendix F. Using the Message File Conversion Utility” on page 123 and *VisualAge Generator Server Guide for MVS, VSE, and VM* for more information on the message file conversion utility.

Message tables eliminate the need for maintaining special message files and databases and allow messages to be displayed with or without the message identifier.

- The message file identifier in the program specification must be at least 3 characters long.
- Under CSP/AD and CSP/AE, a four-character message file ID in the application specification resulted in the fourth character being moved into the message prefix at run time to assist in identifying the application. This does not occur in VisualAge Generator. As a result, each program must do one of the following:
 - Have its own message table
 - Drop the use of the fourth character, if you want to use a single table
- When you print a program, only the item list for the message table is printed; the table contents are not printed. To print the table contents, print the message table by itself.

Transferring Control in the CICS Environment

For the CICS environment, the X'FFFFFFFF' fullword at the end of the parameter list is consistent with the CSP/AE CALL interface. For generated C++ programs, the length of the COMMAREA does not include this fullword unless /ENDCOMMAREA is specified as a C++ generation option for the calling program. If the generation option /ENDCOMMAREA was specified and the parameter format COMMPTR is in effect, the length specified for COMMAREA on the EXEC CICS LINK command is automatically increased by 4 bytes. Under certain conditions, CICS passes a copy of the COMMAREA to the called program. Specify /ENDCOMMAREA to ensure that the X'FFFFFFFF' fullword is included when a copy of the COMMAREA is made.

National Language Codes

The NLS code has been extended to 3 characters. The first two characters define the base language (for example, EN is for English). The third character can either represent natural continuation or identifies variation, script, or dialect. The NLS codes are as follows:

Table 13. Language Support Suffixes

Description	V3.3	VisualAge Generator
US English	E	ENU
Simplified Chinese	C	CHS
German (Germany)	G	DEU
German (Switzerland)	W	DES
Japanese	J	JPN
Korean	K	KOR
Portuguese (Brazilian)	P	PTB
Spanish	n/a	ESP

CSP/AE Invocation Parameters

CSP/AE invocation parameters have been replaced with generation options, installation options, or are no longer supported.

Table 14 shows how CSP/AE invocation parameters are supported when using generated C++ applications.

Table 14. Invocation Parameter Migration for CSP/AE

Parameter	Corresponding Function in VisualAge Generator
P=yyyy	You can specify an alternate print destination by adding an entry for EZEPRINT in the resource association part. The program can dynamically change the print destination by using the special function word EZEDESTP.
NLS=n	The NLS code for VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX is specified at run time with the EZERNLS environment variable.
SEG	The program switches between segmented and nonsegmented mode by setting EZESEGM.
DMODE=S D A	The DMODE parameter has been replaced by the /DBMS generation option. You choose whether DB2 or Oracle SQL statements are generated using /DBMS. DB2 SQL or Oracle SQL is imbedded directly into the generated C++ program.
RT=zzzz	The RT option is specified at generation time through the /RT generation option. The user no longer has to press attention interrupt to go on to the next transaction; the transaction is started automatically.

Table 14. Invocation Parameter Migration for CSP/AE (continued)

Parameter	Corresponding Function in VisualAge Generator
TSMS	The /WORKDB generation option specifies whether auxiliary or main temporary storage is used for saving working storage across segments.
FFFF	The /ENDCOMMAREA generation option specifies whether the end of list indicator is used.

Appendix D. Host Environments to VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX

This appendix discusses general considerations for migrating applications from host environments supported by Cross System Product or VisualAge Generator to VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX. In addition, refer to the compatibility consideration information in the *VisualAge Generator Programmer's Reference*.

The ASCII collating sequence is different from the EBCDIC collating sequence. Therefore, VisualAge Generator functions that are affected by collating sequence might give different results on the workstation than on the host. For example, because table contents are converted to ASCII, a range check might give a different result on the workstation than on the host. Similarly, the record retrieved after doing a SET record SCAN might be different than on the host.

VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX handles abends differently than CSP/370RS or VisualAge Generator Server for MVS, VSE, and VM. For example, if a VisualAge Generator program calls a non-VisualAge Generator program, which in turn calls another VisualAge Generator program, then the non-VisualAge Generator program must handle the abend.

DL/I database access is not supported by VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX.

VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX supports CICS OS/2 and COBOL file access.

Conversion of Host Data Files

Data files used by programs that are run on the host can be converted to a format that can be used on the workstation.

A conversion utility is provided to convert data formatted for use on the host to data formatted for use on the workstation. For information on the conversion utility, refer to the *VisualAge Generator Server Guide for OS/2, AIX, Windows NT, and HP-UX* and see the online help topic on converting files.

SQL Considerations

The following are considerations you must be aware of when migrating SQL applications to the workstation:

- The ASCII collating sequence, which is normally used for databases on the workstation, is different from the EBCDIC collating sequence, which is normally used for databases on host systems.
- Several differences exist between the SQL support provided by DB2/2, DB2, DB2/VSE, and SQL/DS. These are listed in the *Formal Register of Extensions and Differences in SQL* (SC26-3316). The following are examples of these differences:
 - DB2/2 Version 1.0 does not support concatenation of character columns using the !! operator.
 - DB2/2 Version 1.0 can require a table qualifier on column names that appear in two different tables when both tables are referenced in a SQL statement.

If you are a CSP 3.3 VM SQL user and wish to migrate the database environments, APAR PN76454 (CSP/AD 3.3) and PN75537 (CSP/AE 3.3) provide you with VM static module support. With VM static module support, you can move your host SQL data to the database in other systems through SQL DRDA protocol.

Appendix E. Migrating to VisualAge Generator with TeamConnection

This appendix lists considerations that apply when migrating to VisualAge Generator with TeamConnection. Refer to the *VisualAge Generator Generation Guide* and the *IBM VisualAge TeamConnection Enterprise Server User's Guide* (SC34-4499) for more information.

Source Code

When you use the ENVY/TeamConnection bridge to transfer VisualAge Generator parts from the ENVY library manager to TeamConnection, consider the following:

- When using TeamConnection, each global data item creates both a data item and a shared data element in the TeamConnection family database. Review the use of global data items to determine whether any of them should be changed to local data items. For example, keeping data items in SQL row records as global makes it easier to change all uses of the data item if the SQL table changes. However, if a data item is only used in one working storage record, it might be better to change it into a local data item.
- Review the ENVY applications to remove any parts that are no longer used. Eliminating obsolete parts helps reduce the size of the TeamConnection family database and also helps improve performance.

Linkage Table Part

Linkage table parts must be included in the configuration map that is used by the TeamConnection build process. When a linkage table is modified, all programs that were built with the linkage table are built again by TeamConnection. However, ITF only allows one linkage table to be specified in the **Preferences** notebook on the **VAGen - Test Linkage** page. Therefore, if you are using a single linkage table, you might want to consider keeping the single linkage table for the purposes of ITF testing, but splitting the linkage table into smaller tables for the purposes of doing builds with TeamConnection.

Resource Association Part

Resource association parts used in the generation of a program (by specifying the `/RESOURCE` generation option) must be contained in an ENVY application that is included in the configuration map used by the TeamConnection generation build process. If a resource association part is used to specify file characteristics for many programs, any change to the resource association part will cause all those programs to become out of date in TeamConnection. For this reason, you might want to split resource association parts that are used during generation of many programs into separate resource association parts for each program. Resource association parts that are only used at runtime need not be split in this manner.

Appendix F. Using the Message File Conversion Utility

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. Depending on your migration needs and the execution environments you choose, you can use either of the following two utilities to perform a message file conversion:

- CSP/AE 3.3 Message File Conversion Utility (If you have not installed VisualAge Generator Server for MVS, VSE, and VM, see “Using CSP/AE 3.3 Message File Conversion Utility”.)
- VisualAge Generator Message File Conversion Utility (If you have already installed VisualAge Generator Server for MVS, VSE, and VM, see “Using VisualAge Generator Message File Conversion Utility” on page 127.)

Using CSP/AE 3.3 Message File Conversion Utility

The MSGUTIL utility contains two Cross System Product applications, MSGUTIL and MSGMPR. The source for MSGUTIL and MSGMPR is shipped in a single ESF file (MSGUTIL ESF for VM or DCAMSGU ESF for MVS and VSE). You must import the ESF file for these applications using CSP/AD 3.3 and generate them into an ALF.

Before running the MSGUTIL utility from CSP/AE 3.3, make sure you have installed the required APARs and have allocated the data sets as follows:

For VM users:

1. Make sure APAR PN61895 has been installed.
2. Allocate file MSGFILE to an existing VSAM message file.
3. Allocate file MSGTBLE to a CMS file to receive the external source format. The file must have 80-byte fixed-length records.
4. Allocate file MSGOPTS to a CMS file to include the conversion parameter statement. The file must have 80-byte fixed-length records. Figure 11 on page 125 shows the syntax of the conversion parameter statements.
5. Modify the CSPAEBAT EXEC shipped with CSP/AE 3.3 to invoke the MSGUTIL utility.

6. Start the MSGUTIL utility from CSP/AE 3.3.

For MVS users:

1. Make sure APAR PN61895 has been installed.
2. Allocate file MSGFILE to an existing VSAM message file.
3. Allocate file MSGTBLE to a QSAM file to receive the external source format. The file must have 80-byte fixed-length records.
4. Allocate file MSGOPTS to a QSAM file to include the conversion parameter statement. The file must have 80-byte fixed-length records. Figure 11 on page 125 shows the syntax of the conversion parameter statements.
5. Start the message utility MSGUTIL from CSP/AE 3.3 as a MVS batch job. Figure 12 on page 126 shows the sample JCL.

For VSE users:

1. Make sure APAR PQ04253 has been installed.
2. Allocate file MSGFILE to an existing VSAM message file.
3. Allocate file MSGTBLE to a VSAM ESDS file to receive the external source format. The file must have 80-byte fixed-length records.
4. Allocate file MSGOPTS to a VSAM ESDS file to include the conversion parameter statement. The file must have 80-byte fixed-length records. Figure 11 on page 125 shows the syntax of the conversion parameter statements.
5. Start the message utility MSGUTIL from CSP/AE 3.3 as a VSE batch job. Figure 13 on page 127 shows the sample JCL.
6. Use VSE/ESA Interactive Interface Utility to transfer to the workstation the message tables in VSAM ESDS created from step 4.
Refer to the *VSE/ESA Programming and Workstation Guide* for detailed instructions on how to move VSAM files from VSE to the workstation.

Return Codes

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.

Message Table Conversion Parameters

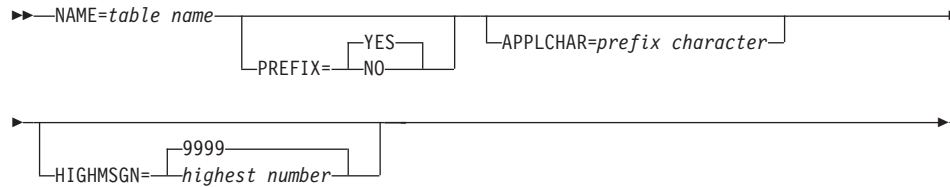


Figure 11. Message Table Conversion Parameters

NAME	<p>Specifies the name of the table member to be created for VisualAge Generator Developer.</p> <p>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 application specifications, and ccc is a language code.</p> <p>Valid language codes are as follows:</p> <p>ENU U.S. English</p> <p>ENP Uppercase English</p> <p>CHS Simplified Chinese</p> <p>DEU German</p> <p>DES Swiss German</p> <p>ESP Spanish</p> <p>JPN Japanese (Kanji)</p> <p>KOR Korean</p> <p>PTB Brazilian Portuguese</p>
PREFIX	<p>Specifies whether the prefix should be included in the message text.</p> <p>YES Specifies that the prefix should be included in the message text. The default is YES.</p> <p>NO Specifies that the prefix should not be included in the message text.</p>
APPLCHAR	<p>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</p>

files, this character corresponds to the fourth character in the Message File field of the CSP application 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.

Sample MVS JCL for CSP 3.3 Message File Conversion Utility

```
//MSGUTIL JOB
//*-----
//*
//*      SAMPLE JCL FOR RUNNING THE MSGUTIL UTILITY
//*
//*-----
//ONE EXEC PGM=DCGBINIT
//*-----
//*      CSP/AE FILES - LOADLIB AND MESSAGE FILE
//*-----
//STEPLIB DD DSN=CSP.V3R3M0.AELOAD,DISP=SHR
//DCADZGD DD DSN=CSP.V3R3M0.DZGMSG,DISP=SHR
//*-----
//* User's ALF which contains MSGUTIL utility.
//*-----
//UMSGALF DD DSN=SSHIEH.FZERSAM,DISP=SHR
//*-----
//*      PRINT FILES
//*-----
//EZEPRINT DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=654,BLKSIZE=658)
//SYSPRINT DD SYSOUT=A
//*-----
//*      EXECUTION PARMS
//*-----
//DCAPARM DD *
A=UMSGALF.MSGUTIL
/*
//SYSOUT DD SYSOUT=A,DCB=(RECFM=FB,BLKSIZE=133)
//SYSPRINT DD SYSOUT=A
//* MSGFILE - VSAM INPUT FILE DD STATEMENT
//MSGFILE DD DSN=xxxxx.MSGFILE.RRDS,DISP=SHR
//* MSGTBLE - QSAM OUTPUT FILE DD STATEMENT
//MSGTBLE DD DSN=SSHIEH.MSGTBLE,DISP=SHR
//MSGOPTS DD *
NAME=MSGENU PREFIX=YES APPLCHAR=A HIGHMSGN=9999
/*
//
```

Figure 12. Sample JCL for Running MSGUTIL for MVS Batch

Sample VSE JCL for CSP 3.3 Message File Conversion Utility

```
* $$ JOB JNM=MSGCRUN,CLASS=6,DISP=D,NTFY=(CARVM3,SSHIEH)
* $$ LST CLASS=A,DEST=(CARVM3,SSHIEH)
* $$ PUN CLASS=A,DEST=(CARVM3,SSHIEH)
* Message file conversion utility
// JOB MSGCRUN 410 DELETE MISC
// ASSGN SYS005,SYSLST
// OPTION PARTDUMP
// DLBL SHVSAMU,'SHVSAMU',,VSAM
// DLBL CS2VSMU,'CS2VSMU',,VSAM
* -----
* Existing message file - Replace with your dataset name
* -----
// DLBL MSGFILE,'CSP.V3R3M0.EZEMSG.C',,VSAM,CAT=SHVSAMU
* -----
* File to receive definition in external source format
* -----
// DLBL MSGTBLE,'SSHIEH.MSGTBLE',,VSAM,CAT=CS2VSMU
* -----
* Option file
* Valid options are:
* name = Represents names of the table member
* prefix = Whether or not a message prefix is desired
* applchar= Application specific character for insert
* highmsgn= Highest message number to convert
* -----
// DLBL MSGOPTS,'SSHIEH.MSGOPTS',,VSAM,CAT=CS2VSMU
* -----
* Run the Message Conversion Utility CSP/AE application (MSGUTIL)
* -----
// DLBL UMSGALF, 'SSHIEH.UMSGALF',,VSAM,CAT=CS2VSMU
// DLBL DCADZGD, 'CSP.V3R3M0.DZGMSG.C',,VSAM,CAT=SHVSAMU
// LIBDEF PHASE,SEARCH=(PRD3.CSP330M,PRD3.CSP330A,PRD3.CSPCOMM)
// EXEC PGM=DCBYINIT,SIZE=64K
A=UMSGALF.MSGUTIL U=SSHIEH
F=EZEPRINT S=005
;;
/*
/&
* $$ E0J
```

Figure 13. Sample JCL for Running MSGUTIL for VSE Batch

Using VisualAge Generator Message File Conversion Utility

The DZGMSGC utility is shipped with VisualAge Generator Server for MVS, VSE, and VM. Use this utility if you have installed VisualAge Generator Server in these environments.

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 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. Allocate MSGOPTS to a sequential file to include the conversion parameter statement. Figure 11 on page 125 shows the syntax of the conversion parameter statement.
4. Start the message utility, DZGMSGC.

For MVS, VSE and VM sample execution code, refer to the *VisualAge Generator Server Guide for MVS, VSE, and VM*.

Appendix G. Host Generation Option Changes

Table 15 shows option name changes from the host generation options files to the VisualAge Generator generation facility.

Table 15. Host Generation Option Changes

Cross System Product Options	VisualAge Generator Options
CHECKTYP(0 1 2)	/CHECKTYPE=LOW ALL NONE
COMMLVL	/COMMENTLEVEL
CONVT	/CONTABLE
EXECJCL	/RUNFILE /NORUNFILE
FFFF	/ENDCOMMAREA /NOENDCOMMAREA
INIADDWS	/INITADDWS /NOINITADDWS
MAPS(ALL NONE)	/GENMAPS /NOGENMAPS /GENHELPMAPS /NOGENHELPMAPS
MATH(CSP) or MATH(COBOL)	/MATH=CSPA /MATH=COBOL
MFSFEAT	/MFSIGNORE /NOMFSIGNORE
NLS	/TARGNLS
PREPJCL	/PREPFILE /NOPREPFILE
PRINT(YES NO)	/LISTING /NOLISTING /LISTINGONERROR
TRANSID('txid')	/TRANSID= primary,restart
TABLES(ALL NONE)	/GENTABLES /NOGENTABLES
USERID	/PROJECTID

Index

A

- accessing same table, applications 47
- application
 - definition 4
 - storage requirements 94
- association, system resource 78

C

- client/server processing support 80
- coexistence
 - CSP/2AD 1.2 to VisualAge Generator 33, 59
 - CSP/AD programmable workstation feature 3.3 to VisualAge Generator 33, 59
 - with CSP/AE applications 95
- configuration map
 - definition 4
- control information
 - definition 4
 - general 6
 - storing in ENVY 7
- CONVERSE process option 85, 112
- conversion utility, message file 123
- converting, data files 119
- cooperative processing
 - support 80
 - VisualAge Generator Server for MVS, VSE, and VM to VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX 119
- Cross System Product
 - control information 6
- CSP/2AD
 - date edit mask 81
 - processing statement extensions 80
 - set extended attributes 81
 - system resource association 78
 - transactional processing 77
- CSP/2AD 1.2 MSLs 59
- CSP/2RS 1.1 to VisualAge Generator Server for OS/2, AIX, Windows NT, and HP-UX 65
- CSP/2RS file migration
 - considerations 65
 - CSP/2RS file migration
 - considerations, table of duplicate entries 66
- CSP/370AD
 - application load file 94
 - generating applications 62, 67
 - generating map groups 62, 67
 - generating tables 62, 67
- CSP/370RS
 - 1.1 and VisualAge Generator, accessing same table 47
 - 1.1 to VisualAge Generator Server for MVS, VSE, and VM 45
 - 1.1 to VisualAge Generator Server for MVS, VSE, and VM, installation considerations 45
 - 1.1 to VisualAge Generator Server for MVS, VSE, and VM, preparation procedures 45
 - 2.1 to VisualAge Generator Server for MVS, VSE, and VM 61
 - 2.1 to VisualAge Generator Server for MVS, VSE, and VM, procedures for 62
 - V2.1 to VisualAge Generator Server for MVS, VSE, and VM, installation considerations 61
- CSP/AD
 - 3.2.2 to VisualAge Generator, resource associations 19, 30
 - 3.2.2 to VisualAge Generator Developer 25
 - 3.3, generation output libraries 42
 - 3.3, interfacing to PL/I 43
 - 3.3, options file 37
 - 3.3, reserved word file 40
 - 3.3, to VisualAge Generator Developer 37
 - 3.3 to VisualAge Generator, resource associations 19, 30
 - 3.3 to VisualAge Generator Developer 25
- CSP/AD, CSP/AD 3.3 Programmable Workstation Feature or CSP/2AD 1.2 to VisualAge Generator Developer 33, 59

- CSP/AD, generation options 19, 30
- CSP/AD 3.2.2 or 3.3 13, 25
- CSP/AE
 - coexistence with 95
 - invocation parameters 98
- CSP/AE, invocation parameters, C++ 117
- CSP/AE 3.3 message file conversion utility, using 123
- CSP/AE 3.3 to VisualAge Generator Server for MVS, VSE, and VM 21, 35, 83
- cursor with hold 76

D

- data values 91
- data values, C++ 115
- date edit mask for map and data item definition
 - for CSP/2AD 81
 - for CSP/370AD 81
- DCBINIT routine 104
- DCBRINIT routine 104
- defining PCBs, for the work database 47
- defining PSBs 47
- definition limitations when migrating, CSP/370AD 4.1 52
- DL/I,
 - DL/I database support when migrating 115
- DL/I, DL/I database support when migrating 91
- DL/I support 119

E

- ELATSRST routine 104
- ENVY
 - characteristics 3
 - concepts 4
 - functional organization 8
 - part types 5
 - storing control information 6
 - storing parts 6
- error routines 47
- EZE special function words 86, 112
- EZECONCT option
 - for CICS OS/2 73
 - for MVS 81
- EZEDLPCB 81

F

field edits, variable 85
field edits, variable, C++ 112
field lengths, C++ 112
field lengths, COBOL 85
file support when migrating 88, 113
functional organization 4, 8

G

generated tables 47
generation output libraries 42

I

installation considerations, migrating
from CSP/370RS 2.1 61
installation considerations,
VisualAge Generator Server for
MVS, VSE, and VM 45
interfacing to PL/I 43

J

JCL templates 40

L

language codes 98
language codes, C++ 116
language-dependent variables 93
language differences
Cross System Product to
VisualAge Generator 73
CSP/AD 3.3 to VisualAge
Generator 1.0 77
cursor with hold 76
EZECONCT option for CICS
OS/2 73
from CSP/370AD 4.1 to
VisualAge Generator 1.0 73
language element
compatibility considerations 84
compatibility considerations,
application load file 94
compatibility considerations,
CONVERSE process option 85
compatibility considerations,
DL/I database support 91
compatibility considerations, EZE
special function words 86
compatibility considerations, file
support 88
compatibility considerations,
language-dependent variables
93
compatibility considerations,
mapping support 84

language element (*continued*)
compatibility considerations,
relational database (SQL)
support 90
compatibility considerations,
transferring among applications
87
language element,
compatibility considerations, C++
111
compatibility considerations,
CONVERSE process option
112
compatibility considerations,
DL/I database support 115
compatibility considerations, EZE
special function words 112
compatibility considerations, file
support 113
compatibility considerations,
mapping support, C++ 111
compatibility considerations,
relational database (SQL)
support 114
compatibility considerations,
transferring among applications
113
linkage types 20, 31

M

map support 84, 119
map support, C++ 111
message file conversion utility, using
123
migrating
CSP/AD 3.3 to VisualAge
Generator Developer 37
from CSP/370AD 4.1 53
templates 56
from CSP/370AD 4.1, language
differences 73
from CSP/370AD 4.1 to
VisualAge Generator Developer
51
from CSP/370RS 1.1 to VisualAge
Generator Server for MVS, VSE,
and VM, upward compatibility
46
from CSP/370RS 2.1, upward
compatibility 62
from CSP/AD 3.3, JCL templates
40
from CSP/AD 3.3, language
differences 77
from CSP/AD 3.3, plan definition
43

migrating (*continued*)

from CSP/AE 3.2.2 to C++
VisualAge Generator Server for
OS/2, AIX, Windows NT, and
HP-UX 111
from CSP/AE 3.3 21, 35, 83
from CSP/AE 3.3 to C++
VisualAge Generator Server for
OS/2, AIX, Windows NT, and
HP-UX 111
SETGEN information from
CSP/AD 19, 31
SQL applications 120
migrating an MSL, from VSE/ESA to
the workstation 13, 25
migrating considerations
from CSP/AE 3.3 to VisualAge
Generator Server for MVS, VSE,
and VM 21, 35, 83
VisualAge Generator Server for
MVS, VSE, and VM language
element compatibility
considerations 84
VisualAge Generator Server for
MVS, VSE, and VM member
names 83
migrating considerations,
VisualAge Generator Server for
MVS, VSE, and VM language
element, C++ 111
VisualAge Generator Server for
MVS, VSE, and VM part names,
C++ 111
migration considerations
data values 91
language-dependent variables
93
user message files when
migrating 93
migration considerations,
data values, C++ 115
user message files when
migrating 115
migration utilities
generation option changes 129
MSL
concatenation 8
correspondence to member type
5
correspondence to VAGen part
class 5
definition 3
functional organization 8
member types 5
names of 5

- MSL (*continued*)
 - storing control information 6
 - storing members 6
- MSLs 33
- MVS batch, message file conversion utility 123
- MVS CICS
 - invocation parameter migration 98, 100, 117
 - resource table requirements 97
 - table entries 97
- N**
- national language
 - applications 104
 - codes 98
- national language codes, C++ 116
- O**
- options file 37
- ownership 4
- P**
- PCB, defining for the work database 47
- PCB, passing in EZEDLPCB 81
- PL/I, interfacing to 43
- plan definition 43
- preparation procedures 45
 - procedure
 - for migrating from CSP/370RS 2.1 62
- process, migration 3
- processing statement
 - extensions for CSP/2AD 80
 - extensions for CSP/370AD 80
- PSB, defining 47
- R**
- relational database (SQL) support 90, 114
- reserved word file 40
- resource associations
 - conversion utility 19, 30
- resource table requirements
 - MVS CICS 97
 - VSE CICS 97
- running CSP/370AD V4.1 with CSP/2RS V1.1, upward compatibility 67
- S**
- segment break
 - MVS CICS 104
 - VSE CICS 104
- set extended attributes
 - on CSP/2AD 81
- set extended attributes (*continued*)
 - on CSP/370AD 81
- SQL
 - dynamic table access, on CSP/370AD 81
 - migration considerations 91
 - SQL, migration considerations 115
 - storage, application storage requirements function 94
 - system resource association
 - for CSP/2AD 78
 - for CSP/370AD 78
- T**
- table entries
 - MVS CICS 97
 - VSE CICS 97
- tables for user message files 81
- test facility limitations when migrating 17, 28
- test facility limitations when migrating, CSP/370AD 4.1 52
- testing TSO CICS 17, 28, 52
- transactional processing
 - for CSP/2AD 77
 - for CSP/370AD 77
- transferring among applications 87
- transferring among applications, C++ 113
- transferring application control
 - in the CICS environment 96
- transferring application control, in the CICS environment 116
- types of linkage 20, 31
- U**
- upward compatibility, CSP/370AD V4.1 with CSP/2RS V1.1 67
- upward compatibility when migrating from CSP/370RS 2.1 62
- user message files when migrating 93, 115
- utilities, message file conversion, MVS batch 123
- V**
- variable field edits 85
- variable field edits, C++ 112
- VisualAge Generator message file conversion utility, using 127
- VisualAge Generator Server for MVS, VSE, and VM
 - generating applications 46
 - generating map groups 46
 - generating tables 46
 - member names 83
- VisualAge Generator Server for MVS, VSE, and VM, language element compatibility considerations, C++ 111
- part names, C++ 111
- VM, invocation parameter migration 102
- VSE CICS
 - resource table requirements 97
 - table entries 97



File Number: S370/OS-50



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SH23-0244-01

