

VisualAge Generator



Server Guide for AS/400

Version 4.5

VisualAge Generator



Server Guide for AS/400

Version 4.5

Note

Before using this document, read the general information under “Notices” on page v.

First Edition (September 2000)

This edition applies to the following licensed programs:

- IBM VisualAge Generator Developer for OS/2 and Windows NT Version 4.5
- IBM VisualAge Generator Server for OS/2, AIX, Windows NT, HP-UX, and Solaris Version 4.5
- IBM VisualAge Generator Server for AS/400 Version 4 Release 4
- 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) 445-9269. Faxes should be sent Attn: Publications, 3rd floor.

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.

IBM welcomes your comments. You can send your comments in any one of the following methods:

Electronically, using the online reader comment form at the address listed below. Be sure to include your entire network address if you wish a reply.

- <http://www.ibm.com/software/ad/visgen>

By mail to the following address:

IBM Corporation, Attn: Information Development, Department G7IA Building 503, P.O. Box 12195, Research Triangle Park, NC 27709-2195.

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 1989, 1998, 2000. All rights reserved.**

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

Contents

Notices	v	Preparing a DB2/400 application	19
Trademarks	vii	Chapter 4. Running a generated application in OS/400	21
Terminology used in this document	ix	Making VisualAge Generator Server for AS/400, COBOL, and generated modules available	21
Terminology differences between Java and Smalltalk	x	Establishing a library list for a job.	23
About this document.	xi	Running VisualAge Generator applications under OS/400	23
Who should use this document.	xi	Passing arguments and declaring parameters	23
How this document is organized	xi	VisualAge Generator run unit concept	25
Documentation provided with VisualAge Generator	xiii	Using activation groups with run units	25
OS/400 publications	xv	Using tables with run units	26
Chapter 1. Installing and configuring VisualAge Generator Server for AS/400	1	Chapter 5. General considerations for VisualAge Generator Server for AS/400	27
Installing VisualAge Generator Server for AS/400	1	Using data description specifications generated by VisualAge Generator	27
Objects created or replaced during installation	1	Modifying the generated DDS information	27
Customizing VisualAge Generator for multiple users	2	Restrictions on logical files	28
OS/400 library and file setup	2	Changing DDS member types	28
Setting up an OS/400 job description	3	Application run-time considerations	28
OS/400 run-time CL templates	4	Starting and ending commitment control cycles	28
Customizing the preparation templates	4	VisualAge Generator and CSP/AE applications coexisting	29
Reasons for modifying templates	8	Mixing run units and run unit isolation	29
Customizing VisualAge Generator	9	Sharing native database files	30
Chapter 2. Application generation outputs	11	Map group and table isolation	30
Outputs of generation.	11	Message file compatibility	30
Objects generated for applications.	13	Chapter 6. Moving prepared applications to other OS/400 systems	33
Objects generated for tables.	13	Moving prepared applications to another OS/400 system	33
Objects generated for map groups.	14	Moving workstation code that is part of a VisualAge Generator application to an OS/400 system	33
Objects generated for all member types	14	Maintaining backup copies of production libraries	34
Objects generated for application and message table members	15	Chapter 7. Generator messages	35
Chapter 3. Preparing generated applications	17		
Running the Preparation CL	18		
Analyzing the preparation listing	18		

Chapter 8. VisualAge Generator Server for AS/400 messages	39
Escape messages	39
Diagnostic and informational messages	40

Chapter 9. Diagnosing problems during run time	53
---	-----------

Debugging VisualAge Generator	54
OS/400 First Failure Data Capture component	54

Index	57
--------------	-----------

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.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact 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:

- AD/Cycle
- AIX
- AS/400
- CICS
- CICS Operating System/2
- CICS OS/2
- CICS/ESA
- CICS/MVS
- CICS/VM
- CICS/VSE
- Client Access/400
- COBOL/370
- COBOL/400
- DB2
- DB2/2
- DB2/400
- IBM
- IBMLink
- ILE
- IMS
- IMS/ESA
- Language Environment
- MVS
- MVS/ESA
- Operating System/2
- Operating System/400
- OS/2
- OS/400
- SAA
- SQL/DS
- SQL/400
- VisualAge
- VisualGen
- VM/ESA
- VSE/ESA
- WebSphere

The following are trademarks of other companies:

Micro Focus COBOL	Micro Focus Limited
HP-UX	Hewlett-Packard Company

Microsoft, Windows, Windows NT, the Windows 95 logo, and the Windows 98 logo are trademarks or registered trademarks of Microsoft Corporation.

Solaris, Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

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 for VSE/ESA, CICS/ESA, CICS for OS/2, CICS for AIX, CICS for Windows NT, and CICS for Solaris.
- CICS for Windows NT refers to IBM TXSeries for Windows NT Version 4.2.
- CICS for AIX refers to IBM TXSeries for AIX Version 4.2.
- CICS for Solaris refers to IBM WebSphere Enterprise Edition Version 3.0.
- 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” and “CICS region” correspond to the following:
 - CICS for MVS/ESA region
 - IMS region
 - CICS for VSE/ESA partition
 - CICS for OS/2 system
 - CICS for AIX system
 - CICS for Windows NT system
 - CICS for Solaris 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 for 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.
- DB2/2 applies to DB2/2 Version 2.1 or later, and DB2 Universal Database (UDB) for OS/2 Version 5.
- DB2/6000 applies to DB2/6000 Version 2.1 or later, and DB2 Universal Database (UDB) for AIX Version 5.
- Windows applies to Windows 95, Windows 98, Windows NT, and Windows 2000.
- Unless a specific version of Windows NT is referenced, statements regarding Windows NT also apply to Windows 2000.

Terminology differences between Java and Smalltalk

VisualAge Generator Developer can be installed as a feature of VisualAge for Java or VisualAge Smalltalk. Where appropriate, the documentation uses terminology that is specific to Java or Smalltalk. But where the information is specific to VisualAge Generator and virtually the same for both environments, the Java/Smalltalk term is used.

Table 1. Terminology differences between Java and Smalltalk

Java term	Combined Java/Smalltalk term	Smalltalk term
Project	Project/Configuration map	Configuration map
Package	Package/Application	Application
Workspace	Workspace/Image	Image
Beans palette	Beans/Parts palette	Parts palette
Bean	Visual part or bean	Visual part
Repository	Repository/ENVY library	ENVY library manager
Options	Options/Preferences	Preferences

About this document

This document provides information on installing and configuring VisualAge Generator Host Services for OS/400*, preparing and running VisualAge Generator-generated COBOL applications in the OS/400 environment, and diagnosing problems. Explanations and user responses for the run-time messages are also provided.

Who should use this document

This document is intended for anyone responsible for installing, configuring, developing, and running applications on VisualAge Generator Server for AS/400.

How this document is organized

The chapters in this document are designed to help you with:

- Installing and configuring VisualAge Generator Server for AS/400
- Preparing and running applications in the OS/400 environment
- Developing applications with OS/400 specific considerations
- Moving applications to other systems
- Diagnosing problems

Documentation provided with VisualAge Generator

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

- Printed and separately ordered using the individual form 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.
- HTML files (.htm) on the product CD-ROM and from the VisualAge Generator web page (<http://www.ibm.com/software/ad/visgen>).

The following books are shipped with the VisualAge Generator Developer CD. Updates are available from the VisualAge Generator Web page.

- *VisualAge Generator Getting Started* (GH23-0258-01) ^{1,2}
- *VisualAge Generator Installation Guide* (GH23-0257-01) ^{1,2}
- *Introducing VisualAge Generator Templates* (GH23-0272-01) ^{2,3}

The following books are shipped in PDF and HTML formats on the VisualAge Generator CD. Updates are available from the VisualAge Generator Web page. Selected books are available in print as indicated.

- *VisualAge Generator Client/Server Communications Guide* (SH23-0261-01) ^{1, 2}
- *VisualAge Generator Design Guide* (SH23-0264-00) ¹
- *VisualAge Generator Generation Guide* (SH23-0263-01) ¹
- *VisualAge Generator Messages and Problem Determination Guide* (GH23-0260-01) ¹
- *VisualAge Generator Programmer's Reference* (SH23-0262-01) ¹
- *VisualAge Generator Migration Guide* (SH23-0267-00) ¹
- *VisualAge Generator Server Guide for Workstation Platforms* (SH23-0266-01) ^{1,4}
- *VisualAge Generator System Development Guide* (SG24-5467-00) ²
- *VisualAge Generator User's Guide* (SH23-0268-01) ^{1, 2}
- *VisualAge Generator Web Transaction Development Guide* (SH23-0281-00) ¹

The following documents are available in printed form for VisualAge Generator Server for AS/400 and VisualAge Generator Server for MVS, VSE, and VM:

- *VisualAge Generator Server Guide for AS/400* (SH23-0280-00) ²
- *VisualAge Generator Server Guide for MVS, VSE, and VM* (SH23-0256-00) ²

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

2. These documents are available in hardcopy format.

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

4. This document is included when you order the VisualAge Generator Server product CD.

The following information is also available for VisualAge Generator:

- *VisualAge Generator External Source Format Reference* (SH23-0265-01)
- *Migrating Cross System Product Applications to VisualAge Generator* (SH23-0244-01)
- *VisualAge Generator Templates V4.5 Standard Functions—User's Guide* (SH23-0269-01)^{2, 3}

OS/400 publications

The following documents contain information related to running applications in the OS/400 environment:

- *AS/400 ILE Concepts V4R4*, SC41-5606
- *AS/400 Programming Reference Summary V4R4*, SX41-5720
- *AS/400 System Operation for New Users V3R1*, SC41-3200
- *AS/400 System Operation V3R6*, SC41-4203
- *CSP/AE Reference*, SC41-3613
- *Data Management*, SC41-5710
- *DB2 for AS/400 Database Programming*, SC41-5701
- *DB2 for AS/400 Distributed Database Programming*, SC41-5702
- *DB2 for AS/400 SQL Programming*, SC41-5611
- *DB2 for AS/400 SQL Reference*, SC41-5612
- *DB2/400 Database - An Overview*, SC41-3700
- *DDS Reference*, SC41-5712
- *Getting Started*, SC41-5601
- *Getting Started on the AS/400 System*, SC41-3204
- *ILE COBOL for AS/400 Programmer's Guide*, SC09-2540
- *ILE COBOL for AS/400 Reference*, SC09-2539
- *OS/400 Backup and Recovery V4R4*, SC41-5304
- *OS/400 CL Programming V4R4*, SC41-5721
- *OS/400 Program and CL Command APIs V4R4*, SC41-5870
- *OS/400 Security - Reference V4R4*, SC41-5302
- *OS/400 Server Concepts and Administration*, SC41-3740
- *OS/400 Work Management V4R4*, SC41-5306
- *Performance Tools V4R2*, SC41-5340
- *Publications Reference*, SC41-3003
- *Web Programming Guide V4R4*, GC41-5435

Chapter 1. Installing and configuring VisualAge Generator Server for AS/400

This chapter contains general information on the installation and customization of VisualAge Generator Server for AS/400 on the host and VisualAge Generator on the workstation.

Installing VisualAge Generator Server for AS/400

VisualAge Generator Server for AS/400 (program number 5769-VG1) is available on a CD. To install VisualAge Generator Server for AS/400 Version 4.4 on the host, enter the following command:

```
RSTLICPGM LICPGM(5769-VG1) DEV(OPTnn)
```

where *nn* is the two-digit number of the device where the CD-ROM is loaded.

The estimated time to run the RSTLICPGM LICPGM(5769-VG1) DEV(OPTnn) command is 15 minutes.

Objects created or replaced during installation

The following list provides a general description of the objects created or replaced during the installation process:

Table 2. Objects Created or Replaced during the Installation Process

Object and library name	Type	Description
QVGEN QSYS	*LIB	The primary library for those system objects that constitute VisualAge Generator Server for AS/400.
QVGN* QGPL	*FILE	Contains various database and source files, some of which are used during development and run time of VisualAge Generator applications. See "Customizing VisualAge Generator for multiple users" on page 2 for a description of these files.
/QVGN	DIR	The Integrated File System directory for run-time performance tuning of VisualAge Generator applications. Java parts for the gateway server are placed in this directory.
/QVGEN	DIR	The Integrated File System directory where Java parts for the gateway server reside.

Customizing VisualAge Generator for multiple users

Although the VisualAge Generator Server for AS/400 and VisualAge Generator products are installed as ready to use, you should read this section for customization considerations that can better meet your project development needs and to assist in understanding and integrating development tasks with your OS/400 system management.

The following is a list of the customization steps:

1. Create OS/400 user libraries to contain generation output and application executables.
2. Copy to user libraries OS/400 files that contain generated output for each developer or project.
3. Tailor templates that the application generator uses to produce OS/400 job streams to prepare generated applications and associated parts.
4. Create OS/400 job descriptions that are used by the OS/400 job streams.
5. If you are using client server support of VisualAge Generator to call non-VisualAge Generator servers, upload the file QVGNRNCL.CLR from the VisualAge Generator Developer directory to the AS/400 and compile the program using the CRTCLPGM command. Place the program in every library that you have non-VisualAge Generator server programs.

OS/400 library and file setup

The outputs from generation are, by default, placed into files in the QGPL library on the OS/400 system. The files are created in the QGPL library during installation of VisualAge Generator Server for AS/400. The output files from generation are as follows:

Table 3. Generation Output Files

File name	Description
QVGNCBLS	ILE COBOL application programs and print services
QVGNCLS	Preparation CL and run-time CL
QVGNMSGs	Message CL
QVGNMAPG	VisualAge Generator MapGroup
QVGNDDSS	Data Description Specifications (DDS)
QVGNTAB	VisualAge Generator Data Tables
QVGNJOB	Submitted preparation job streams
QVGNPCF	Generation parts control file

To preserve the original installation-defined definition of the QVGN* files in the QGPL library and to avoid member name collisions when multiple application developers are using the same host OS/400 system, it is highly recommended that you copy the QVGN* files from the QGPL library to specific application developer user libraries. You can name these user libraries any name you choose, and you can set up as many as necessary to meet your needs.

The application developer user library corresponds to the VisualAge Generator option /DESTLIB= value specified during application generation. The default for this value is QGPL. After application generation and preparation is complete, you can leave the files and programs where they are, move them, or copy them to any other library for run time.

To create libraries for multiple developers or projects, do the following:

1. Type the following command on the command line:

```
CRTLIB xxxxxx
```

where xxxxxx is the library name.

2. Create a duplicate of the VisualAge Generator files in the new library by typing the command:

```
WRKOBJ OBJ(QGPL/QVGN*) OBJTYPE(*FILE)
```

3. Place *option=3* (copy) next to each file, then type the command:

```
TOLIB(XXXX)
```

Setting up an OS/400 job description

An OS/400 job description is required for preparation before you can submit batch jobs. It is recommended that each VisualAge Generator application developer have a job description on the OS/400. You can create a job description with the CRTJOB command. Name the job description the same as the application developer's LAN ID. If the application developer does not have a LAN ID, then either set the VisualAge Generator environment variable EZERUSRID to the job description name or the template EFK14PBJ must be modified.

Specify the application developer's OS/400 user ID in the job description, do not use the default *RQD. Set message logging to *SECLVL. The job description must be in a library named the same as the userid. The system default library list (*SYSVAL) does not contain user libraries.

You can customize the job descriptions and templates in whatever way you want to meet your project requirements as long as you do not use the default *RQD in the job descriptions.

An example of creating a job description is as follows:

```

CRTJOB                                +
  USER(userid)                        +
  LOG(4 00 *SECLVL)                   +
  JOB  (userid/userid)                +
  INLLIBL (destlib QVGEN otherlibs) +

```

where *userid* is the user ID, *destlib* is the library where the generation output files should be stored, and *otherlib* is the library you can use for whatever you want.

OS/400 run-time CL templates

For all called OS/400 applications, VisualAge Generator creates a run-time CL that is called when the VisualAge Generator application is used as a server. The file is generated with the name *mbrname.clr* and uploaded as *mbrname_R* to the QVGNCLS file on the OS/400.

Each server application runs in a prestarted job with the job description specified in the QUSER profile. The run-time CL is used to add libraries to the library list so that programs and data files used by the server can be found. The templates used to create the run-time command file are as follows:

EFK24EBC

This is the run-time prologue template. This template establishes CL program entry and global error monitoring during client/server run time.

EFK24EEC

This is the run-time epilog template. This template adds the library that the server application is in to the library list. If this is the first server program in a client run unit, the CL command sets up commitment control. Add additional libraries to the beginning of the CL for data and programs in libraries that are different from the server program.

Customizing the preparation templates

VisualAge Generator can be customized to meet the needs of any application development project. All of the customization is in the area of application generation and encompasses the use of generation options and application preparation templates.

Generation options are keywords named or referred to on the command to generate a VisualAge Generator application member. Refer to the *VisualAge Generator Generation Guide* document for more information about generation options.

Preparation templates are files that represent a unique segment of instructions for larger generation output files. These output files manage the transfer, compile, and bind of application members.

Preparation templates overview

The VisualAge Generator Developer default templates produce OS/400 control language (CL) programs to manage the task of compiling and binding ILE COBOL programs. The CL programs have an extension of CLP and CLJ in the generation output directory. When the CL programs are transferred to OS/400, they are placed in the files QVGNCLS and QVGNJOB in the library named on the /DESTLIB generation option. Each generated member for QVGNCLS or QVGNJOB has the same name as the program part from which it was generated. The submitted batch job (.CLJ) compiles and calls the .CLP file that prepares the generated programs. When the CL program is run, generated VisualAge Generator programs are prepared for run time.

All of the templates have the file-name extension .TPL and are originally installed in the directory C:\Program Files\Vast\bin for the Smalltalk version of VisualAge Generator or C:\IBMJava\IDE\program for the Java version. The following are the templates for the OS/400 environment and their functions:

EFK24PBC

Establishes CL program entry and global error monitoring during preparation of one VisualAge Generator part type

EFK24PCL

Provides CL to compile and bind ILE COBOL/400 when the VisualAge Generator part is a CALLED (non-SQL) program type

EFK24PEC

Provides an epilogue to the CL preparation program, such as general purpose error handling, during preparation of one VisualAge Generator part type

EFK24PMN

Provides CL to compile and bind ILE COBOL/400 when the VisualAge Generator part is a MAIN (non-SQL) program type

EFK24PPM

Provides CL to compile and bind ILE COBOL/400 when the VisualAge Generator part is a map group containing printer maps (mapping services program)

EFK24PSC

Provides CL to DB2/400 pre-compile, COBOL compile, and bind ILE COBOL/400 when the VisualAge Generator part is a CALLED program type containing SQL statements

EFK24PSM

Provides CL to DB2/400 pre-compile, COBOL, compile, and ILE COBOL/400 when the VisualAge Generator part is a MAIN program type containing SQL statements

EFK24WCL

Provides CL to compile and bind ILE COBOL/400 when the VisualAge Generator part is a Web transaction (non-SQL) program type

EFK24WSC

Provides CL to DB2/400 pre-compile, COBOL compile, and bind ILE COBOL/400 when the VisualAge Generator part is a Web transaction program type containing SQL statements

The message file CL used for the OS/400 environment has the file-name extension .MSG. CL for the message file is similar to the preparation for VisualAge Generator program and map group print map part types. The message file preparation CL that is produced from templates prepares a VisualAge Generator message table part when started. During the preparation phase, message table parts are converted to OS/400 message file objects to be used by VisualAge Generator programs. CL message templates are used for the OS/400 environment as follows:

EFK24TCM

Provides CL to create the message file

EFK24TAM

Provides CL to add a message to the message file

EFK24TEM

Provides an epilogue to the CL preparation program, such as general purpose error handling, during preparation of one VisualAge Generator part type

The run-time file CL, used for client/server programs in the OS/400 environment, has the file extension .CLR. During the preparation phase, the run-time CL source is copied to the file QVGNCLS in the destination library, with the name of prgname_R. It is then compiled.

EFK24EBC

Provides the CL to add libraries to the client/server job and to start commitment control if this is the first server program called by a client. Customize this template to add any libraries that the server program needs for data or programs.

EFK24EEC

Provides an epilogue to the run-time CL to handle errors that occur.

Batch jobs used for the OS/400 environment have the file extension .CLJ. OS/400 batch jobs are used to create and then run the preparation CL programs previously defined by the CL templates. The batch job stream is produced from the following templates:

EFK24PBJ

Establishes the database batch job entry. This template must contain the //BCHJOB command. This template is similar to the MVS JOB card template. This template can also contain job setup commands, such as library list alterations and job value changes.

Note: The generation option /JOB CARD= value can be used to override this template name with one of your own choosing, enabling easy customization to batch job parameters that control the job definition.

EFK24TMJ

Provides CL commands to create the message CL preparation program, and then runs it. This template is used when the VisualAge Generator part is a message file.

EFK24PEJ

Provides CL commands to create the program CL preparation program, and then runs it. This template is used when the VisualAge Generator part is a program or a print map.

Modifying templates

Templates can be modified to meet installation requirements for preparing and running applications. These modifications are usually done by a system administrator or project leader.

Templates are used to generate files used in preparing members for run time. The template files required for generation are shipped with the VisualAge Generator application generator.

The file-name extension for template files is .TPL. These sample template files are placed in the directory C:\Program Files\Vast\TEMPLATE during installation. The installation procedure updates the OS/2 environment variable PATH to include the C:\Program Files\Vast\TEMPLATE directory.

If the /TEMPLATES generation option is specified, VisualAge Generator searches for templates in the specified directory. The specified directory must contain copies of all the templates. If the /TEMPLATES generation option is not specified, VisualAge Generator searches for templates in the directories listed by the PATH environment variable of the process currently running the EZERGEN command. If you want to modify the templates, create a new directory. Then, consider the following:

- If you intend to use the /TEMPLATES generation option to make your modifications, copy all the templates from the C:\Program Files\Vast\TEMPLATE directory into the new directory and apply your changes.
- If you prefer to use the DPATH environment variable, copy only those templates you want to change and add this new directory to the DPATH environment variable in the OS/2 process that starts the EZERGEN command.

The templates contain symbolic parameters that are indicated by the percent sign (%). Values can be defined and substituted in place of these symbolic parameters dynamically. The substitution values are derived from any of the following sources:

- Definitions of the members being generated
- Generation options
- /SYMPARM values

User-defined symbolic parameters can also be inserted in the template. The values for user-defined symbols are specified at generation time using the /SYMPARM generation option. When using symbolic parameters in a template, ensure that the generated statement does not exceed column 71 when the largest possible values are substituted for the symbols. Statements longer than 71 columns in host templates are truncated. Also make sure that your changes result in the generation of valid syntax. The generation facility does not check the syntax.

Refer to the *VisualAge Generator Generation Guide* under “Symbolic parameters” for more information about symbolic parameters.

Reasons for modifying templates

You should not need to modify templates often. However, if a template needs to be changed, your system administrator or project leader usually makes the change. Some reasons why templates might need to be modified include the following:

- Conforming to your system or project application development file naming conventions
- Changing the preparation process commands run on OS/400; either changing the order of, adding, or removing commands
- Changing command parameter values; either hardcoded or add VisualAge Generator symbolic parameters, predefined or user-defined
- Changing the library list setup at different stages of the preparation batch job
- Overriding objects during the preparation batch job execution, such as the VisualAge Generator COBOL compiler options copybook member, VGNCBLOP

- Changing the COMMIT parameter value on the CRTSQLCBLI command to control the commitment control level for SQL statements in an application
- Changing the ownership or granted authority to created objects
- Changing the parameters on the //BCHJOB command, such as using a different job description or changing the message logging. It might be better in this case to create a new template with a name of your choosing and use the /JOB CARD= VisualAge Generator generation option to include it instead of the VisualAge Generator- supplied template (EFK24PBJ.TPL) for the purpose of using //BCHJOB command variations.

For more information on OS/400 batch job management, refer to the *AS/400 Work Management* document. For more information on CL programming, refer to the *AS/400 CL Programming* document.

Customizing VisualAge Generator

If you have installed VisualAge Generator Server for AS/400 in a language other than the following languages, VisualAge Generator Server for AS/400 creates an abbreviated conversion table in the file QVGNSTCB of the QVGEN library.

Suffix Language

ENU	U.S. English
ENP	Uppercase English
CHS	Simplified Chinese
DEU	German
DES	Swiss German
ESP	Spanish
JPN	Japanese (Katakana)
KOR	Korean
PTB	Brazilian Portuguese

Chapter 2. Application generation outputs

This chapter provides an overview of the output files from application generation to OS/400. VisualAge Generator prepares the output files before running your applications. The format and content of the output files are in COBOL/400, control language (CL) source, and structured binary streams. Each output file is described in Table 4.

Outputs of generation

After you generate an application, a number of objects must be transferred to the OS/400 host system as members in various OS/400 physical files. On the OS/400 host system, these members must be prepared before the application can be run.

Table 4 provides information about the types of files produced by generation, including the following:

- Type of object produced
- Physical file name where the object is written as a member
- How the member name of the object is derived
- Whether production is controlled by a generation option
- Whether the object can be modified after generation is performed

Refer to the *VisualAge Generator Generation Guide* document for more information on controlling and modifying generation and preparation of OS/400 objects.

Table 4. Objects transferred to an OS/400 host by the VisualAge Generator preparation utility

File type	Physical file name	PF member name and generated file name	Generation option	Modifiable
Objects generated for applications				
Application COBOL/400 program	QVGNBLS	Application name <i>applname.CBL</i>	None	No
Run-time CL	QVGNCLS	Application name <i>applname.CLR</i>	None	Yes
Objects generated for tables				
Table Binary Image	QVGNTAB	Table name <i>tablename.TAB</i>	/GENTABLES	No
Objects generated for map groups				

Table 4. Objects transferred to an OS/400 host by the VisualAge Generator preparation utility (continued)

File type	Physical file name	PF member name and generated file name	Generation option	Modifiable
Print services program (See note 3)	QVGNCBLS	Map group name <i>mapgroupname.CBL</i>	/GENMAPS /GENHELPMAPS	No
Map group module (See note 4)	QVGNMAPG	Map group name <i>mapgroupnameFM.FMT</i>	/GENMAPS /GENHELPMAPS	No
Objects generated for all member types (applications, tables, map groups)				
Listing file	(See note 6)	Member specified when generation was requested <i>mbrname.LST</i>	/LISTING /LISTING ONERROR	None
Parts control file (See note 5)	QVGNPCF	Member specified when generation was requested <i>mbrname.PCF</i>	None	No
Preparation Command File	(See note 6)	Member specified when generation was requested <i>mbrname.PRP</i>	/PREPFILE	No
Preparation CL	QVGNCLS	Application name <i>applname.CLP</i>	/PREPFILE	Yes
Preparation job stream CLJ	QVGNJOB	Application name <i>applname.CLJ</i>	/PREPFILE	Yes
Objects generated for message tables				
Message file CL	QVGNMSG	Member specified when generation was requested <i>tablename.MSG</i>	/GENTABLES	Yes
Message file CLJ	QVGNCLJ	Message table name <i>tablename.CLJ</i>	/GENTABLES	Yes
Objects generated for file creation				
Data definition specification (DDS)	QVGNDDSS	File name as specified in VisualAge Generator record definitions <i>filename.DDS</i>	/CREATEDDS	Yes

Table 4. Objects transferred to an OS/400 host by the VisualAge Generator preparation utility (continued)

File type	Physical file name	PF member name and generated file name	Generation option	Modifiable
Notes:				
1. The generator produces ILE COBOL/400 for the OS/400 environments.				
2. Generated application, table, and map group objects are environment dependent. All objects are generated for one environment and cannot be used in another environment.				
3. This object is produced only if the map group contains print maps.				
4. This object is produced only if the map group contains terminal maps.				
5. The parts control file is no longer generated. Version 3.1 and earlier versions of VisualAge Generator Developer generated this file. One parts control file is created for a GENERATE subcommand. The parts control file has the same name as the member specified on the GENERATE subcommand. If any map groups or tables are generated with an application, the parts control file for the application lists all parts generated for the application and all parts generated for the associated map groups and tables.				
6. These files are not transferred to the host. The OS/2 file name is the same as the name of the member specified on the GENERATE subcommand.				

Objects generated for applications

The follow sections describes the objects generated for applications.

Application COBOL/400 program

The generated application is a COBOL/400 program that contains the following:

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

Run-time CL

The run-time CL sets commitment control and adds libraries to the OS/400 library list when a server application or Web transaction runs.

Data definition specification (DDS)

The generator produces OS/400 data definition specifications (DDS) to create instances of OS/400 physical and logical files that the application uses. The DDS produced is the result of the *indexed*, *relative*, *serial*, and *alternate specification* record types used within the application member being generated. The generation option /CREATEDDS enables the production of the DDS output type. The preparation command file uploads the DDS files to the host system, but the command file does not manage processing beyond that point.

Objects generated for tables

The following section describes the objects generated for tables.

Table binary image file

The table binary image file contains the run-time VisualAge Generator table member contents as defined by VisualAge Generator Developer. The table contents are already converted to the code page of the target run-time environment. The table contents are formatted to an application defined structure (possibly containing various data types) and the contents are treated as binary data. You might not be able to view the contents outside the scope of VisualAge Generator Developer and utilities. The generation option /GENTABLES enables the production of the table binary image files.

Objects generated for map groups

The following section describes the objects generated for map groups.

Map group format module

The map group format module is a generated structure that describes the map layout for terminal maps in the map group. The generator builds the structure as a binary image file converted to the code page of the target system. This object is produced when you specify the /GENMAPS or the /GENHELPMAPS generation options and the application has defined terminal maps in the map group.

Objects generated for all member types

The following objects are generated for applications, tables, and map groups.

Listing file

A source listing of the member being generated and its associates is produced if you specify one of the print options (/LISTING or /LISTINGONERROR) during generation. The listing includes any messages that were issued and options that were in effect during generation.

Parts control file

The parts control file, generated in earlier versions of VisualAge Generator Developer (see Table 4 on page 11), is a list of all objects produced by the generator the last time generation was performed for the application, table, or map group with the same name as the parts control file member. This list provides a record of generated objects.

Preparation command file

A preparation command file is generated when the /PREPFILE generation option is specified. The preparation command file produced by a GENERATE subcommand contains logic to prepare the generated members. If the generated application includes tables and maps, the logic for preparing them is contained in the preparation command file.

The preparation command file runs on the workstation and contains commands used to transfer the source files and preparation CL file to the

target host system. One preparation command file is created for each GENERATE subcommand that was issued.

Objects generated for application and message table members

The following objects are generated for applications and message tables.

Preparation CL for OS/400 environments

The generator produces control language (CL) statements to compile and bind generated applications. The CL statements produced depends on the nature of the program:

- SQL applications require DB2/400 precompile.
- CL programs must compile generated applications.

The CL statements are produced from model CL templates that you can modify to enforce source file naming conventions or preparation procedures. Refer to “Customizing the preparation templates” on page 4 for more information on preparation templates.

The preparation command file on the workstation transfers the preparation CL to the host system and submits it to the batch subsystem.

Preparation job stream file

The OS/400 job stream file contains the commands needed to compile and run the preparation CL program. The CL program compiles the VisualAge Generator applications.

Message file CL

The generator produces CL statements to create or update OS/400 native message file objects (*MSGF type). VisualAge Generator message table members are implemented as message files on OS/400. The CL is produced from model CL templates that you can modify to control OS/400 message description attributes and error handling during run time of the CL statements. Refer to the *VisualAge Generator Generation Guide* document for more information.

The generation option /GENTABLES enables the production of the message file since the origin of the data is the VisualAge Generator message table. The preparation command file uploads the message file CL to the host system and submits it to the batch subsystem.

Chapter 3. Preparing generated applications

This chapter describes the preparation procedures for preparing an application to run in the OS/400 environment. It is assumed that you are familiar with jobs and work management on OS/400 and with transferring files and submitting remote commands using ftp. If you are not familiar with these tasks, see “OS/400 publications” on page xv for a list of documents that cover these tasks.

After VisualAge Generator generates one (or an associated set) of its source members, the generated output is transferred to the target run-time system and prepared to run. For OS/400, VisualAge Generator produces a set of specific command files to manage the application preparation process for each VisualAge Generator source member that is scheduled for generation.

Application preparation includes the following steps:

- Transferring generation output files to the host system
- Compiling the various application COBOL/400 source files
- Binding the compiled modules with VisualAge Generator Server for AS/400
- Creating and updating application run-time support objects, such as message files and tables

These subprocess preparation steps are common to the third-generation language (3GL) method of application development. VisualAge Generator relieves the application developer from having to manage and individually perform these preparation steps. The only task left for the application developer to perform is to verify and test the resulting run-time objects.

VisualAge Generator cannot predict or provide every possible run-time scenario an application developer might design. However, VisualAge Generator does provide the flexibility that enables the application developer to customize the preparation process, both before and after VisualAge Generator produces the command files that control the preparation process. For automating the preparation process and development process control, it is recommended that you customize your preparation process *before* VisualAge Generator produces the command files. Refer to “Customizing the preparation templates” on page 4 for information on the preparation templates and how to modify the templates. The remainder of this chapter covers specific subprocess preparation steps, such as the following:

- Running the CL
- Analyzing the listings and run-time targets
- Recovering from errors (analyze, fix, and restart the preparation process) that may occur during the preparation process

Running the Preparation CL

VisualAge Generator Developer submits the preparation CL file built by the generator to the OS/400 batch subsystem. If the preparation job ends with an error that can be corrected without requiring the application to be generated again, you can do one of the following:

- Use the VisualAge Generator Developer PREPARE subcommand to submit the preparation CL file to the batch subsystem again
- Issue the OS/400 SBMDBJOB (Submit Database Job) command to submit the preparation job stream CLJ file

Analyzing the preparation listing

This section describes the procedures for analyzing the preparation listing for error messages related to the generation of the CL.

The first place to start when verifying the preparation listing is the generator messages and the message log. If the messages indicate that the generation output files did not transfer correctly, determine the error. After the error is corrected, issue the VisualAge Generator generator PREPARE subcommand to start the preparation process again (this process includes transferring files). Issuing the PREPARE subcommand to start the preparation process is much faster than issuing the GENERATE subcommand and reproducing the generation output files. Refer to the *VisualAge Generator Generation Guide* document for more information on producing and locating the preparation listing and using the PREPARE subcommand.

Note: Messages that begin with the prefix EZE are the VisualAge Generator messages and are listed in “Chapter 7. Generator messages” on page 35. Other messages are produced by ftp.

If the generation output files transferred correctly but the OS/400 SBMDBJOB command failed or a command within the submitted job failed, analyze the type of error on OS/400 by examining job logs and spooled output files, such as the compile and bind listings.

To view the job log and the listings produced by the preparation phase, use the Work with Submitted Jobs command

```
WRKSBMJOB SBMFROM(*USER)
```

for jobs submitted by your user ID or the command

```
WRKJOB JOB(jobname) OPTIONS(*SELECT)
```

where *jobname* is the //BCHJOB JOB(*jobname*) value from the submitted database job stream file on OS/2 (genmember.CLJ) or on OS/400 (destlib/QVGNJOB(genmember)).

If you receive a message that the job you are requesting is no longer on the system, then any spooled output (listing files) that might have been produced has either been printed or deleted. To verify that the submitted job is producing a job log, check the submitted database job stream file for either the //BCHJOB LOG(4 00 *SECLVL) value or the //BCHJOB LOG(*JOBID). In the latter case, verify the job description named on the //BCHJOB LOG(*JOBID) parameter for the same information.

It is recommended that you set up each VisualAge Generator application developer with a user ID on OS/400 and a personal job description object for the developer to reference when submitting batch jobs such as the preparation job stream. The installed VisualAge Generator generator batch job header template (EFK24PBJ.TPL) is set up to use a job description with the same name as the value of VisualAge Generator OS/2 environment variable EZEUSRID in a library with the same name as the EZEUSRID variable.

If errors concerning a preparation batch job on OS/400 can be corrected so that no new generation files are needed, and all the most recent generation output files have already been transferred to OS/400, you can submit the preparation batch job via the SBMDBJOB command again as in the following example:

```
SBMDBJOB FILE(destlib/QVGNJOB) MBR(genmbr)
```

Using the preparation batch job via the SBMDBJOB command requires less time to restart a preparation batch job than using the GENERATE or PREPARE generator subcommands.

Refer to the *AS/400 Work Management* document, SC41-3306, for more information on OS/400 batch jobs and batch job control.

Preparing a DB2/400 application

When you specify an SQL table name in VisualAge Generator Developer, you can enter the table name on the **Table Specification** window using either the SQL naming convention of collection.tablename or the OS/400 SYSTEM naming convention of collection/tablename. Whichever format you use as the standard to qualify table names, tailor the OPTION parameter on the CRTSQLCBLI command in the VisualAge Generator DB2/400 application templates (EFK24PSM.TPL and EFK24PSC.TPL) to be *SQL (collection.tablename) or *SYS (collection/tablename). In the template, the default naming convention is *SQL.

When you tailor the VisualAge Generator DB2/400 application templates (EFK24PSM.TPL and EFK24PSC.TPL), ensure that the *APOSTSQL and *QUOTE values are part of the OPTION parameter. The *APOSTSQL and *QUOTE values are required for VisualAge Generator DB2/400 applications.

Chapter 4. Running a generated application in OS/400

This chapter describes the information required to run VisualAge Generator applications on an OS/400 system.

Making VisualAge Generator Server for AS/400, COBOL, and generated modules available

The setup tasks that are required to run VisualAge Generator applications on OS/400 are simpler than with other run-time environments because no run-time setup control programs are produced, other than the application program itself.

VisualAge Generator Server for AS/400 and the generated COBOL applications use the run-time job library list (*LIBL) to resolve all named object references

The library list must be set up by the application programmer, system administrator, or VisualAge Generator application developer before starting the application. To aid in the library setup, Table 5 on page 22 lists the names and types of objects that VisualAge Generator might use while running in the OS/400 environment. VisualAge Generator searches for these objects dynamically when the application is running by scanning the libraries named in the library list. It is recommended that the installation library, QVGEN, be added to the library list of the end user when running VisualAge Generator applications.

VisualAge Generator uses the first object it finds that matches the target name in the libraries named in the library list. This first-found object is used in all cases of object resolution except for objects of *FILE type. In this case, VisualAge Generator uses the first member it finds that matches the target name, after the first member have been qualified with the correct file name. Multiple files with the same name might exist in the libraries named in the library list. VisualAge Generator checks each library file until it finds the first instance of the member name.

Table 5. Names and types of objects used by VisualAge Generator at run time

Object and library name	Type	Description
QVGN* QVGEN	*PGM *SRVPGM	VisualAge Generator Server for AS/400 program and service program objects.
OVGNMMSGF QVGEN	*MSGF	VisualAge Generator Server for AS/400 product message file.
QVGNMAPG userlib	*FILE	Members of this file contain the generated applications 5250 map groups. Members are named for the map group it contains.
QVGNTAB userlib	*FILE	Members of this file contain the generated applications tables. Members are named for the table it contains.
QVGNPRNT QVGEN	*FILE	This is the standard printer device file for application use of the EZEPRINT file. Usually, all jobs on the system share one of these objects.
QVGMAP QVGEN	*FILE	This standard display device file is used for interactive applications when they display maps. Usually, all jobs on the system share one of these objects.
mmmmnls userlib	*MSGF	A specific application's message table, where <i>mmmm</i> is the message table prefix as defined to the application, and <i>nls</i> is the value of generation option /TARGNLS when the application was generated.
calltarg userlib	*PGM	Any target of the CALL, XFER, or DXFR statements coded within an application.
filetarg userlib	*FILE	Any file named on record definitions used by VisualAge Generator process options within an application.

Note: The designated library *userlib* in Table 5 indicates that the object is in a library named by the application developer at the time the application was developed.

The two exceptions to using the library list to resolve object references by running applications are as follows:

- When VisualAge Generator application tables and map groups reside in the OS/400 IFS file system for improved run-time performance.
- When either:
 - The object (table or view) was explicitly qualified when an SQL record was defined during development

- The object was implicitly qualified when the application using the record was compiled.

In either case, SQL object resolution is independent of the library list.

Establishing a library list for a job

You can establish a library list for a job in several ways, but each method involves using an OS/400 system command. You can also mix the methods. The initial OS/400 library list is contained in the job description referenced by the user profile. For more information about the following OS/400 system commands, refer to the *AS/400 CL Programming* document or enter the command at a command line entry on OS/400 and request command prompting.

ADDLIBLE

Adds a library list entry

CHGCURLIB

Changes the current library

CHGSYSLIBL

Changes the system library list

Running VisualAge Generator applications under OS/400

To run a VisualAge Generator application in the OS/400 environment, call the *PGM application object just as you would any other *PGM object on OS/400. You can run VisualAge Generator applications from menus, commands, command lines, or interlanguage program calls. This applies to MAIN and CALLED VisualAge Generator applications. Examples of how to run a VisualAge Generator application from an OS/400 command line follow:

- To call an application without the use of arguments, use the following format:
`CALL applname`
- To call a CALLED application that expects a parameter declared as CHA 16, use the following format:
`CALL applname ('char arg literal')`
- To call a CALLED application that expects two parameters declared as CHA 15 and PACK 15,5, use the following format:
`CALL applname ('char arg literal' 1234)`

Passing arguments and declaring parameters

Arguments are passed to VisualAge Generator applications in the same way that other OS/400 *PGM objects pass and receive arguments. Both MAIN and CALLED VisualAge Generator applications can receive arguments; however, MAIN applications have a more fixed argument format, and CALLED

applications offer parameter definition diversity. Arguments are always passed by reference. That is, a pointer to the argument is passed.

MAIN applications can optionally receive the following two arguments:

- The application working storage record
- A FIRSTMAP argument

The application working storage record is a structured binary data format that should match or map exactly to the working storage record structure defined in the application for the length of the defined working storage record. If the application working storage record does not match or map to the working storage record, design the application logic to handle uninitialized record fields.

To indicate to the application the amount of data contained in the working storage record argument, a 2-byte length field immediately precedes the record data and is considered part of the argument. The target VisualAge Generator MAIN application accesses only that portion of the record data that is within the length bounds specified by the lesser of the following:

- The length passed with the record data
- The length of the application's defined working storage record

The details of the working storage record are important only if you are using a non-VisualAge Generator or non-CSP/AE application to pass the argument, because the non-VisualAge Generator or non-CSP/AE program establishes the argument storage layout. VisualAge Generator and CSP/AE applications account for all internal structures and data typing where records with the same name are specified on the XFER or DXFR statement and on the working storage of the target application.

The FIRSTMAP argument is a VisualAge Generator-specific format. The FIRSTMAP name is defined to the application during application development. The structure of the argument is a programming interface for VisualAge Generator only.

CALLED applications that are defined with parameters must receive matching arguments when called. Otherwise, you can receive unpredictable results. Do not reference an argument that either has not been received or has been received but with a data type different from its declared matching parameter. Usually, a function check occurs.

For VisualAge Generator CALLED applications, you can declare a maximum of 30 parameters during application development. Parameters can be declared as VisualAge Generator data items, records, or maps.

Record arguments for CALLED application targets do not have the leading 2-byte length field preceding the record data like the working storage record arguments for MAIN application targets.

VisualAge Generator CALLED applications that are called from an OS/400 command line or CL can receive literal numeral arguments if the VisualAge Generator parameters are declared as type PACK, 15 digits, 5 decimals. For more information on the literal data types, refer to the *OS/400 CL Programmer's Guide*

VisualAge Generator run unit concept

VisualAge Generator applications operate in a run unit much like that of ILE COBOL. VisualAge Generator's run unit can be considered a subset of the ILE COBOL run unit because the COBOL run unit might exist before and persist longer than VisualAge Generator's run unit.

VisualAge Generator's run unit is bounded by the first VisualAge Generator application on the OS/400 program call stack for a specific job. Run units are scoped within a single job. As long as a VisualAge Generator application is on the program call stack, a VisualAge Generator run unit is active. Only one VisualAge Generator run unit can be active in a job at any time. This is the most obvious difference between VisualAge Generator run units and ILE COBOL run units.

MAIN or CALLED applications can initiate a VisualAge Generator run unit. VisualAge Generator MAIN applications can exist in a run unit only if it is the initiating application. MAIN applications cannot be called from an application that initiates a VisualAge Generator run unit, even if it is called from a non-VisualAge Generator program while a VisualAge Generator run unit was active.

Using activation groups with run units

VisualAge Generator run units normally correlate on a one-to-one basis with ILE COBOL run units and ILE activation groups. When a VisualAge Generator application initiates or begins a run unit, a named activation group is also initiated. Using a named activation group ensures that all VisualAge Generator applications that run in the job share the same resources in terms of ILE resource management.

If your application system consists of non-VisualAge Generator programs as well as VisualAge Generator applications, you can add your non-VisualAge Generator programs to the named activation group or use a different activation group. Sharing commitment control logical units of work and sharing database file Open Data Paths are important aspects to consider when making your decision. To share either ILE resources, using the same activation

group achieves the above result. Conversely, using different activation groups keeps the ILE resources isolated in terms of VisualAge Generator application use and non-program use.

The ILE activation group name is established when VisualAge Generator applications are in the preparation phase of application development. VisualAge Generator preparation templates, which create OS/400 program objects, name the activation group in the OS/400 command template for the CRTPGM command. The keyword is ACTGRP and the default is ACTGRP(QVGN). The preparation templates that use the CRTPGM command follow:

- EFK14PCL.TPL
- EFK14PMN.TPL
- EFK14PSC.TPL
- EFK14PSM.TPL
- EFK24WCL.TPL
- EFK24WSE.TPL

See “Customizing the preparation templates” on page 4 for information on changing templates.

Using tables with run units

The resource type of data tables used for a VisualAge Generator run unit is specific to VisualAge Generator and is not a function of ILE or OS/400. VisualAge Generator tables are static entities, which are intended for quick table search programming tasks. The tables can be shared among applications in the same run unit. For these reasons, the tables are loaded into storage from a database file. Usually, table in-storage scoping ceases with the run unit in which the tables were loaded. However, the following two factors associated with VisualAge Generator table management warrants table in-storage scoping to extend beyond the VisualAge Generator run unit:

- The time necessary to acquire memory, to read a table from a database file, and to load the table into memory
- The need to retain modified table data for use by another run unit, thereby serving as a quick application-to-application communication method

VisualAge Generator table in-storage scoping extends beyond the VisualAge Generator run unit when the table is defined with the table attribute tag RESIDENT and the run unit is ended by the DXFR statement without the non-CSP option. This means that the application eventually enters the run unit again. When it does, it wants the RESIDENT tables of the previous run unit to remain available in storage. The DXFR statement is the non-committal method of ending a run unit. All other methods close the RESIDENT table, all other tables, and all other VisualAge Generator resources.

Chapter 5. General considerations for VisualAge Generator Server for AS/400

This chapter describes the general considerations for administering VisualAge Generator Server for AS/400 in the OS/400 environment.

Using data description specifications generated by VisualAge Generator

During generation, VisualAge Generator can generate data description specifications (DDS) information from VisualAge Generator record definitions that are used for file I/O operations.

The DDS information generated by VisualAge Generator is useful only to the OS/400 system administrator or application developer. The system administrator can use the DDS source members, or modified versions of them, to create the files that do not already exist on the OS/400 system. Using the DDS source information to create the files qualifies these files for OS/400 data management functions, such as specifying key fields, unique keys, and logical files.

You are not required to use the DDS source information to create files because VisualAge Generator does not require that the files an application accesses be externally described. VisualAge Generator relies on the record definition, which is built into the *PGM object, for the structure of a record. However, using the DDS information guarantees agreement between the application's view of the record structure and the record data stored on the OS/400 system.

Modifying the generated DDS information

Indexed and indexed alternate specification record organizations might require that you modify the corresponding DDS source member. Adding DDS keywords to the file- and record-level identifiers in the DDS source member is the minimum modification necessary. Table 6 on page 28 shows the DDS keywords and the conditions under which they are required.

You can add other DDS keywords to optimize record retrieval and simplify application logic. For example, logical files can be used to select a subset of physical file records. You can also build your own DDS source member, based on your knowledge of the VisualAge Generator record definitions in the application. In this case, individual field names and field lengths in the DDS source need not match those of the VisualAge Generator record definition. However, the record length and key field length of the VisualAge Generator record definition and the DDS source must be equal.

Refer to the *VisualAge Generator Design Guide* document for more information on VisualAge Generator record organizations that cause DDS information to be generated and the recommended OS/400 file types corresponding to those record organizations.

Table 6. Conditions for using DDS keywords

DDS keyword	Condition
PFILE(<i>pfname</i>)	When using the DDS information to create a logical file, <i>pfname</i> identifies the physical file on which the logical file is based. PFILE is a record-level keyword.
UNIQUE	When the application tests for the UNQ or DUP record I/O error conditions. UNIQUE is a file-level keyword.

Restrictions on logical files

VisualAge Generator supports simple logical files that use only one record format. The DDS source information specifies only one file on the PFILE keyword. For complete details on the DDS keywords and their usages, refer to the *DDS Reference*.

Changing DDS member types

VisualAge Generator creates DDS source members without specifying a member type. To assist you in modifying the DDS source information, change the member type to one of the following:

- PF for a DDS source member describing a physical file
- LF for a DDS source member describing a logical file

Changing the member type to PF or LF enables the Source Entry Utility (SEU) prompting to help you to modify the DDS source member.

Application run-time considerations

The following sections describe the considerations to keep in mind during application run time.

Starting and ending commitment control cycles

To use OS/400 Commitment Control Services for single-system OS/400 applications, you must explicitly start and end a commitment control cycle using the start commitment control (STRCMTCTL) command to start the commitment control and the end commitment control (ENDCMTCTL) command to end the commitment control. VisualAge Generator Server for AS/400 does not implicitly start or end commitment control cycles for single-system OS/400 applications. However, DB2/400 implicitly starts commitment control automatically for applications that use SQL I/O process options. After commitment control is started for the job, both native database I/O and SQL I/O can use the common commitment control OS/400 provides.

For VisualAge Generator client/server applications and Web UI applications, commitment control is started by the run-time CL for the application, or, if it is a client/server application, by the generic run-time CL, QVGNRNCL.

If necessary, you can change the commitment control for an SQL application in the templates EFK24PSC.TPL, EFK24PSM.TPL, and EFK24WSC.TPL. The COMMIT parameter on the CRTSQLCBLI command sets the level of commitment control for SQL statements in an application. The LCKLVL parameter on the STRCMTCTL command does not affect applications that use only SQL I/O operations.

If no commitment control cycle is active and the application attempts to open a file requiring commitment control, the application ends with an error condition. Messages in the job log explain the exact nature of the error. The application ends abnormally under these conditions because it might attempt to explicitly commit changes to a file, but that is possible only with an active commitment control cycle.

For more information on OS/400 commitment control services, refer to the *Data Management* document and the *AS/400 ILE COBOL/400 Programmer's Guide*.

VisualAge Generator and CSP/AE applications coexisting

This section provides information on the coexistence between VisualAge Generator and CSP/AE OS/400 applications on VisualAge Generator Server for AS/400 V3.1 and above.

Mixing run units and run unit isolation

VisualAge Generator application run units and CSP/AE application run units coexist in an OS/400 job with little or no modifications necessary for the application types or the application system. If you have a CSP/AE application system that is being migrated to VisualAge Generator, the VisualAge Generator application *PGM objects can replace those that were previously CSP/AE *PGM objects.

The primary consideration to remember when you migrate applications is the run unit startup and end time (performance), both for VisualAge Generator and CSP/AE. Minimize both the number of times a VisualAge Generator or CSP/AE run unit is started for a job and the number of times run unit boundaries are crossed. A CALL statement from a VisualAge Generator application to a CSP/AE application target and visa versa is an example of crossing a run unit boundary. The XFER and, to some extent, DXFR statements are examples of ending the current run unit and starting another depending on the target program language. See “VisualAge Generator run unit concept” on page 25 for more information on run units.

Sharing native database files

VisualAge Generator applications can share, with CSP/AE applications or any other OS/400 program language, a common open data path (ODP) to native database files, if the files are enabled by the file attribute SHARE(*YES). Use the OS/400 command DSPFD (display file description) to view the SHARE attribute as well as other file attributes. For a menu of file commands, type GO CMDFILE on any OS/400 command line.

Map group and table isolation

VisualAge Generator application map groups and table members are isolated from those of CSP/AE in both storage on OS/400 and their use at run time. CSP/AE uses specific object types on OS/400 that correspond to map groups (*CSPMAP object type) and tables (*CSPTBL). VisualAge Generator uses database files and IFS stream files to contain the run-time image of the map group and table members. Therefore, VisualAge Generator map groups and tables can coexist on an OS/400 system with CSP/AE application members with the same name without a name collision because VisualAge Generator and CSP/AE have different object types and storage methods.

During application run time, the run unit instance of map groups and tables are also isolated. Multiple applications that read and write data to a common table in the form of program-to-program communication must be either VisualAge Generator applications or CSP/AE applications, but not a mixture of application types.

Message file compatibility

VisualAge Generator application message tables are implemented on OS/400 as native message file objects (*MSGF). This implementation is similar to CSP/AE message files implementation. The message files are compatible to some degree without modification. The difference between the two message files is that VisualAge Generator message files do not have a text substitution variable (&1) as the first word of the message text. For example, given a message number of 1 with text of 'Hello VisualAge Generator', VisualAge Generator produces and stores the OS/400 message as follows:

```
'Hello VisualAge Generator'
```

CSP/AE produces and stores the same message as follows:

```
'&1 Hello VisualAge Generator'
```

The substitution variable is a place holder for the message ID, which CSP/AE built when the message was retrieved. By using a substitution variable for a place holder for the message ID, CSP/AE ensures that the message ID is always included as displayed information as shown in the following example:

```
'MYMS0001I Hello VisualAge Generator'
```


VisualAge Generator enables the message ID to be optional, to be included or excluded, when the message text is defined as message table content during application development.

VisualAge Generator applications and CSP/AE applications can use either message format if you want to have one common message file for each type of application (VisualAge Generator or CSP/AE). The message file naming scheme for each application type is different to avoid name collisions.

Depending on the message file format you choose, use the OS/400 override message file (OVRMSGF) command to issue a request to bypass one message file and to access the other message file. For example, if you want to use VisualAge Generator message file format, issue the following command so that the CSP/AE application can link to VisualAge Generator message file format:

```
OVRMSGF MSGF(xxx) TOMSGF(yyyyyyy)
```

where *xxx* is the CSP/AE user message file name and *yyyyyyy* is the VisualAge Generator user message file name.

You would reverse the order of the message files in the above example if you want to use the CSP/AE message file format instead of the VisualAge Generator message file format.

Chapter 6. Moving prepared applications to other OS/400 systems

You can move a prepared application from one OS/400 system to another. For example, you might have the compiler on one host development machine but want to run the application on several production machines.

The OS/400 and VisualAge Generator Server for AS/400 products on the production machine must be at the same maintenance level as, or at a higher level than, the development machine.

You will need to recompile or regenerate the application if your production machine is a level 3.1 or 3.2 and the prepare-application machine is at 3.6 or above, or vice-versa. This is due to the underlying change in machine architecture between OS/400 3.1 and 3.2 and OS/400 3.6 and above. Compiled code from one will not run on the other.

Moving prepared applications to another OS/400 system

After an OS/400 VisualAge Generator application is prepared, you can distribute that application to other architecturally similar OS/400 systems using the following procedures:

1. Use the CRTSAVF FILE(library/filename) command to create a save file.
2. Use the SAVOBJ or SAVLIB command to save all the application parts you want to move to another system in the save file.
3. Transfer the save file to other OS/400 systems by using a communications network or by using physical media such as a tape.
4. Use the RSTOBJ (Restore Object) or RSTLIB (Restore Library) command to store the objects from the save file.

Moving workstation code that is part of a VisualAge Generator application to an OS/400 system

Moving workstation code that is part of a VisualAge Generator client/server application to an OS/400 system can be done using diskettes or shared folders.

To save the workstation code using the OS/400, do the following:

1. Copy the code to a shared folder along with VisualAge Generator GUI run-time support
2. Create a save file using the OS/400 command CRTSAVF

3. Issue the OS/400 command SAVDLO
4. Transfer the save file to another OS/400 system by using a communication network or by using a physical media such as a tape
5. Erase the existing shared folder, if one exists
6. Restore the shared folder using the OS/400 command (RSTDLO) on the production OS/400

Maintaining backup copies of production libraries

Maintaining backup copies of your production libraries can be accomplished by creating a save file, then issue the SAVLIB command. You can copy the save file to tape.

Chapter 7. Generator messages

This chapter explains the most common occurring messages received during the generation of VisualAge Generator applications and application members.

EZE4153E **%02C18 - The operand %01C50 is not valid for the source or target of a FIND or RETR statement**

Explanation: Message tables cannot be referenced within a FIND or RETR statement as an operand for the OS/400 environment.

User Response: Define the FIND or RETR statement again so as not to have a message table as one of the operands.

EZE4154I **Map %01C08 has %02C04 too many variable fields**

Explanation: The map indicated contains too many variable fields or BRIGHT constant fields. BRIGHT constant fields are treated as variable fields. The maximum number of variable fields permitted for an OS/400 display depends on the control unit that the device is connected to and the attributes specified for the fields.

This warning message is issued based on the result of the following formula:

$$\frac{4(\text{total} + B\text{cons}) + M\text{var} + D\text{var}}{3} \leq 126$$

where:

total The total number of variable fields (including the number of MIX and DBCS variable fields)

Bcons The number of BRIGHT constant fields

Mvar The number of MIX variable fields

Dvar The number of DBCS variable fields

User Response: To correct the map, do the following:

- Reduce the number of character variable fields or BRIGHT constant fields by %02D04 and generate the map group again.
- Reduce the number of MIX or DBCS variable fields by 80 percent of %02D04 and generate the map again.
- Reduce the number of variable fields and BRIGHT constant fields by a combination that allows the map to fall within the maximum number of variable fields.

Refer to the formula when you reduce items from both of the following categories:

- Character variable fields and BRIGHT constants
- MIX and DBCS variable fields

An application using this map might fail in an OS/400 environment if the fields are not changed.

EZE4172I **Map %01C08 uses row 1, columns 1-4 and field outlining**

Explanation: The map indicated contains a non-blank character in row 1, column 1, 2, 3, or 4. Maps generated for OS/400 displays should have blanks in row 1, columns 1 through 4, if one or more fields on the map are specified with field outlining. Row 1, column 4 can contain a field attribute.

User Response: Enter the map presentation and remove the data from row 1, columns 1 through 4 or, for a partial map, change the start row and column in the map specification. Alternately, enter field attribute definition and change the field outlining attribute specification to NOUTLINE for every field on the map.

Generate the application and map group again. If this message is ignored, execution of an

application using this map might fail in an OS/400 environment.

EZE4195E %01C18 - Variable length records not allowed for OS/400

Explanation: The application referenced a variable length record. Variable length records are not supported in the OS/400 environment.

User Response: Either define the record again to have fixed length, or use a different record for this application.

EZE4196E %01C18 - Table %02C07 exceeded the maximum size allowed

Explanation: The table indicated exceeded the maximum size for a table on the OS/400 system. The maximum size permitted for table on an OS/400 system cannot exceed 3,000,000 bytes.

User Response: To correct the table, reduce the number of entries in the table and generate the table again.

EZE4202E %01C18 - A row in table %02C07 exceeded the maximum size allowed

Explanation: A row in the table indicated exceeded the maximum size for a row in a table on the OS/400 system. The maximum size permitted for a row within table for an OS/400 system cannot exceed 32,767 bytes.

User Response: To correct the table, reduce the size of the row in the table to fewer than 32,767 bytes and generate the table again.

EZE4208E A problem occurred while opening the transfer file

Explanation: The specified built-in routine that opens the indicated transfer file exist, but it has been used incorrectly. Either:

- Invalid arguments were passed to the routine.
- The program called was not compatible with the language processor.

- More than an implementation-limited number of arguments were passed to the routine.

User Response: Edit the .PRP file and correct the LINEOUT routine call to specify the correct parameters.

EZE4273E A problem occurred while writing to the transfer file

Explanation: The specified built-in routine that writes to the indicated transfer file exist, but it has been used incorrectly. Either:

- Invalid arguments were passed to the routine.
- The program called was not compatible with the language processor.
- More than an implementation-limited number of arguments were passed to the routine.

User Response: Edit the .PRP file and correct the LINEOUT routine call to specify the correct parameters.

EZE4289E A problem occurred while opening the file description file

Explanation: The specified built-in routine that opens the indicated file description file exist, but it has been used incorrectly. Either:

- Invalid arguments were passed to the routine.
- The program called was not compatible with the language processor.
- More than an implementation-limited number of arguments were passed to the routine.

User Response: Edit the .PRP file and correct the LINEOUT routine call to specify the correct parameters.

EZE4358E The problem occurred while writing to the file description file

Explanation: The specified built-in routine that writes to the indicated file description file exist, but it has been used incorrectly. Either:

- Invalid arguments were passed to the routine.
- The program called was not compatible with the language processor.

- More than an implementation-limited number of arguments were passed to the routine.

User Response: Edit the .PRP file and correct the LINEOUT routine call to specify the correct parameters.

EZE4362E A host transfer error occurred for file

Explanation: The preparation process detected an error during file transfer to the host that caused the process to stop.

User Response: Verify that Client Access/400 has been started and the link to the OS/400 is made. Start the preparation process again.

EZE4388I A job submission command was issued for

Explanation: The preparation process requires that the previously transferred preparation control language (CL) be submitted to the host for processing.

User Response: No response is required.

EZE4389E The Client Access/400 program could not complete the remote submit request

Explanation: A severe problem occurred while performing a remote submission request for the preparation control language (CL).

User Response: Try running the remote submit request from the Client Access/400 Main menu under the heading of Submit Host System command.

Chapter 8. VisualAge Generator Server for AS/400 messages

For a list of the messages that can be issued by VisualAge Generator applications during run time, see *VisualAge Generator Messages and Problem Determination Guide*. The server messages can also be viewed online on OS/400 by using the Work with Message Description command WRKMSGD MSGF(QVGEN/OVGNMSGF).

Message text can contain one or more inserts. When the message is displayed, an insert is used to fill in names, constants, return codes, and other information. For example, the format of the message insert might look like this: **&1**

VisualAge Generator applications use the standard OS/400 message handling functions to communicate with calling programs. Diagnostic information is automatically logged by the job log during run time.

Escape messages

These messages are sent by VisualAge Generator applications to the program queue of the calling program as OS/400 ESCAPE type messages. The calling program must monitor these messages to avoid an OS/400 function check.

GEN9001 **VisualAge Generator Server
MAIN shell cannot invoke the
target program %1.**

Explanation: Diagnostic messages preceding this message in the job log explain the nature of the error. In most cases, the application or system programmer will need to adjust your application system to correct the problem.

User Response: Either print the job log or record the messages along with the following:

- - The from program name.
- - The to program name.
- - The instruction numbers.

You can view or print the job log with the DSPJOBLOG command. If no diagnostic messages precede this message, ensure that your job logged all messages by checking the

'Message Logging' or LOG value of your job definition or job description, depending on whether the job is interactive or batch. For interactive jobs, command DSPJOB OPTION(*DFNA) will display the 'Message Logging' value.

Contact your application or system programmer with the information you gathered.

GEN9002 **VisualAge Generator Server
encountered an application error
which caused the run unit to end.**

Explanation: Diagnostic messages preceding this message in the job log explain the nature of the error. In most cases, the application or system programmer will need to adjust your application system to correct the problem.

User Response: Either print the job log or record the messages along with the following:

- - The from program name.
- - The to program name.
- - The instruction numbers.

You can view or print the job log with the DSPJOBLOG command. If no diagnostic messages precede this message, ensure that your job logged all messages by checking the

'Message Logging' or LOG value of your job definition or job description, depending on whether the job is interactive or batch. For interactive jobs, command DSPJOB OPTION(*DFNA) will display the 'Message Logging' value.

Contact your application or system programmer with the information you gathered.

Diagnostic and informational messages

These messages are sent as DIAGNOSTIC or INFORMATIONAL type messages to the program queue of the calling program. These messages are automatically posted in the job log. Programs that call VisualAge Generator applications cannot monitor the activities of these messages. Use the WRKJOB(Work with JOB) command to view the job log. The Message Logging job attribute might filter some or all of these messages in some way. To ensure you get all messages posted in the job log, use a message logging value of LOG(4 00 *SECLVL) in your OS/400 jobs. See the OS/400 commands WRKJOB and CHGJOB (Change Job) for more information.

GEN0002 **A new level of IBM VisualAge Generator Server for OS/400 is required for program %1**

Explanation: The generated COBOL program %1, attempting to run, is not compatible with the installed version of VisualAge Generator Server.

User Response: Contact the system administrator. Version %2 of IBM VisualAge Generator Server for OS/400 should be installed.

GEN0005 **Date entered is not valid for defined date format %1**

Explanation: Data entered into a map field defined with a date edit either does not meet the requirements of the format specification as defined using the map definition facility, or the month or day of the month is not valid.

It is not necessary to enter the separator characters shown in the message, but if they are omitted, enter leading zeros. For example, if the date format is MM/DD/YY, you can enter 070494.

User Response: Enter the date in the format %1.

GEN0009 **Overflow occurred because the target item is too short**

Explanation: The target of a MOVE or arithmetic statement is not large enough to hold the result without truncating significant digits. The value of special function word EZEOVER is 1, which specifies that the application should end if this overflow condition occurs.

User Response: Have the application developer do one of the following:

- Increase the number of nondecimal digits in the target data item
- Define the application logic to handle the overflow condition using the special function words EZEOVER and EZEOVERS.

GEN0014 **REPLACE attempted without preceding UPDATE option on %1.**

Explanation: A REPLACE process option was attempted against a record that has not been successfully read by an UPDATE process. The read for update might have been lost as the result of an EZECOMIT or EZEROLLB action.

This error also occurs if a REPLACE is associated with a specific UPDATE process, but that process was not the one used to select the record.

User Response: Have the application developer run this application with the test facility, tracing for process and statement group flow, to determine the application logic error.

GEN0021 **An error occurred in application %1 on VisualAge Generator statement number %2.**

Explanation: The actual error identifying the problem is explained in messages following this message in the job log.

User Response:

GEN0022 **Map group format member %1 could not be loaded**

Explanation: The map group format member is a generated binary file that contains attributes that describe the format and constant fields for character/text based maps in a map group. Map group format members are stored as members of file %2 and are expected to be found by searching the job's library list.

Diagnostic messages preceding this message in the job log explain the nature of the error.

User Response: Contact your application or system programmer and report the sequence of messages including and preceding this message.

GEN0023 **The table %1 could not be loaded.**

Explanation: A table is a generated binary file that contains application data. Tables are stored as members of file %2 and are expected to be found by searching the job's library list.

Diagnostic messages preceding this message in the job log explain the nature of the error.

User Response: Contact your application or system programmer, and report the sequence of messages including and preceding this message.

GEN0024 **VisualAge Generator conversion table %1 could not be found**

Explanation: Either the name specified on the EZECONV call was not a member of the QVGEN/QVGENSCTB file or the member that was found is not a VisualAge Generator Server conversion table.

User Response: Have the application developer verify that the correct conversion table name was specified in the EZECONV call. If the table name was not correct, then change the VisualAge Generator application and regenerate it. If the table name is correct, verify that the correct conversion table was installed. The conversion table is a member in the file QVGENSCTB in QVGEN library.

GEN0026 **A calculation caused a "maximum value" overflow.**

Explanation: During a calculation in an arithmetic statement, an intermediate result exceeded the maximum value (18 significant digits). This condition also occurs when division by zero occurs. If the EZEORDER switch is set to either 0 or 1, the application ends.

This error only occurs when you specify the NUMOVFL(YES) generation option.

User Response: Have the application developer correct the application logic either to avoid the error or to handle the error using the special function words EZEORDER and EZEORDER.

GEN0027 **The data on character to numeric move is not valid.**

Explanation: The statement in error involves a move from a character to a numeric data item. The character data item contains nonnumeric data.

User Response: Have the application developer change the application to make sure that the source operand contains valid numeric data.

GEN0031 A call to program %1 failed.

Explanation: Called programs are expected to be found by searching the job's library list.

Diagnostic messages preceding this message in the job log will explain the nature of the error.

User Response: Contact your application or system programmer, and report the sequence of messages including and preceding this message.

GEN0033 Call to function %1 returned exception code %2

Explanation: An exception code was returned on a call to the specified function, indicating that one of the arguments passed to the function was invalid. The REPLY option was not coded on the function call, specifying that the program should end if an exception occurs.

Refer to the function description in the VisualAge Generator reference manual for an explanation of the exception code.

The run unit ends.

User Response: The developer should fix the program so that it does not pass invalid arguments to the function.

GEN0034 Application %1 was defined as a MAIN application and cannot be called.

Explanation: The specified application was defined as either a MAIN Transaction application or a MAIN Batch application. It cannot be called by another application or program.

User Response: If using the CALL statement to invoke application %1 is valid, have the application developer define %1 as a CALLED application. If application %1 must remain a MAIN application, then have the application developer use XFER or DXFR statements to invoke it from another MAIN application.

GEN0035 Data type error in input - enter again

Explanation: The data in the first highlighted field is not valid numeric data. The field was defined as numeric.

User Response: Enter only numeric data in this field, or press a bypass edit key to bypass the edit check. In either situation, the application continues.

GEN0036 Input minimum length error - enter again

Explanation: The data in the first highlighted field does not contain enough characters to meet the required minimum length.

User Response: Enter enough characters to meet the required minimum length, or press a bypass edit key to bypass the edit check. In either situation, the application continues.

GEN0037 Input not within defined range - enter again

Explanation: The data in the first highlighted field is not within the range of valid data defined for this item.

User Response: Enter data that conforms to the required range, or press a bypass edit key to bypass the edit check. In either situation, the application continues.

GEN0038 Table edit validity error - enter again

Explanation: The data in the first highlighted field does not meet the table edit requirement defined for the variable field.

User Response: Enter data that conforms to the table edit requirement, or press the bypass edit key to bypass the edit check. In either situation, the application continues.

**GEN0039 Modulus check error on input -
 enter again**

Explanation: The data in the first highlighted field does not meet the modulus check defined for the variable field.

User Response: Enter data that conforms to the modulus check requirements, or press a bypass edit key to bypass the edit check. In either situation, the application continues.

**GEN0040 No input received for required
 field - enter again**

Explanation: No data was typed in the field designated by the cursor. The field is required.

User Response: Enter data in this field, or press a bypass edit key to bypass the edit check. Blanks or nulls will not satisfy the data input requirement for any type of field. In addition, zeros will not satisfy the data input requirement for numeric fields. The application continues.

**GEN0041 A message file prefix was not
 specified for an application:
 EZEMNO = %1, NLS code = %2.**

Explanation: A user message was requested either by setting EZEMNO or by specifying a user message number for a map field edit. A user message table prefix was not defined for the application.

User Response: Have the application developer do one of the following:

- - Add the message table prefix to the application specifications and then generate the application again.
- - Change the application to not set EZEMNO and the generate the application again.
- - Remove the user message number from the map field edit and then generate the map group again.

**GEN0045 Error retrieving user application
 message, EZEMNO = %1, NLS
 code %2. See previous messages.**

Explanation: A user message was requested either by setting EZEMNO or by specifying a user message number for a map field edit. The previous message in the job log explains the reason for the error.

User Response: Most problems occur because the message file or the specific message cannot be found or access to the message file is not authorized. If the application can not find the message file and you know the library name that contains the message file, you can add the library to your library list (ADDLIBLE command). For other problems, contact your system or application programmer.

**GEN0046 Call to printer mapping services
 program %1 failed.**

Explanation: Printer mapping services programs are expected to be found by searching the job's library list.

Diagnostic messages preceding this message in the job log explain the nature of the error.

User Response: Contact your application or system programmer and report the sequence of messages including and preceding this message.

**GEN0050 Number of allowable significant
 digits exceeded - enter again**

Explanation: The user entered data into a numeric field that was defined with decimal places, a sign, currency symbol, or numeric separator edits. The number of significant digits that can be displayed within the editing criteria was exceeded by the input data; the number entered is too large. The number of significant digits cannot exceed the field length, minus the number of decimal places, minus the places required for editing characters.

User Response: Enter a number with fewer significant digits.

GEN0051 **Map %1 was not found in map group %2.**

Explanation: The specified map name is not in the map group.

User Response: Have the application developer generate the map group and the application again.

GEN0057 **DELETE attempted without preceding UPDATE on record %1.**

Explanation: A DELETE process option was attempted for a record not successfully read by an UPDATE process option. The read for update might have been lost as the result of an EZECOMIT or EZEROLLB action.

User Response: Have the application developer run this application with the test facility, tracing for process and statement group flow, to determine the application logic error.

GEN0073 **SQL error, command = %1, SQL code = %2**

Explanation: The SQL database manager returned an error code for an SQL statement. Application processing ends following an SQL request whenever the SQLCODE in the SQL communications area (SQLCA) is not 0, and either of the following is true:

- - No error routine is specified for the process
- - The SQLCODE indicated a hard (terminating) error and the special function word EZEFEC was set to 0, indicating that the application should end on SQL error conditions.

This message is followed by the actual DB2/400 message describing the SQL error code.

User Response: Have the application developer or system programmer determine the cause of the problem from the SQL code and the SQL error information. Either correct the application or the database definition. Refer to the appropriate Database Manager messages and codes manual for information on the SQL code and SQL error information.

GEN0074 **SQL error message: %1**

Explanation: This message accompanies message GEN0073 when an SQL error occurs. It displays the relational database manager description of the error and is repeated as many times as necessary to display the complete description.

User Response: Use the information from this message and GEN0073 to correct the error.

GEN0076 **The data on character to hexadecimal move or compare is not valid.**

Explanation: The current statement involves either a move from a character data item to a hexadecimal data item, or a comparison between a character data item and a hexadecimal data item. The characters in the character data item all must occur in the following set for the move or compare to complete successfully:

a b c d e f A B C D E F 0 1 2 3 4 5 6 7 8 9

One or more of the characters in the character data item is not in this set. This condition causes an application program error.

User Response: Have the application developer change the application to make sure that the character data item contains valid data when the character to hexadecimal move compare operation occurs. To do so, the application developer can use the hexadecimal map edit characteristic to make sure that input from a variable field contains valid characters.

GEN0080 **Hexadecimal data is not valid**

Explanation: The data in the variable field identified by the cursor must be in hexadecimal format. One or more of the characters you entered does not occur in the following set:

a b c d e f A B C D E F 0 1 2 3 4 5 6 7 8 9

User Response: Enter only hexadecimal characters in the variable field. The characters are left-justified and padded with the character zero. Embedded blanks are not allowed.

GEN0086 **%1 - No active SETINQ, SETUPD, or UPDATE**

Explanation: A SCAN, REPLACE, or DELETE process option cannot be performed because a required process has not occurred previously. If a REPLACE process has an associated UPDATE or SETUPD process, the UPDATE or SETUPD process must be active. One of the following conditions must exist:

- - A SCAN process for an SQL row must be preceded by a SETINQ or SETUPD process that has the same process object as the SCAN.
- - A REPLACE process for an SQL row must be preceded either by an UPDATE process for the same row or by a SCAN process for a row that was selected using the SETUPD process.
- - A DELETE process for an SQL row must be preceded either by an UPDATE or SCAN process for the same object.

The SCAN, REPLACE, or DELETE process option must be run in the same application that ran the previous process. All rows selected for scanning or updating are released when the application returns to the calling application.

User Response: Have the application developer modify the application to issue the required process option prior to the failing process.

GEN0093 **An error occurred in application %1, process or group %2.**

Explanation: An error occurred in the specified application and process or statement group. Other information about the error is given in the messages that follow this message.

If a process or statement group is not active, the second insert contains the name of a section in the initialization or ending logic for the generated application.

User Response: Refer to the error messages following this message to determine the cause of the error.

GEN0096 **A mixed data operand is not valid**

Explanation: An operand in a move statement involving a mixed data item contains mixed double-byte and single-byte data that is not valid.

User Response: Have the application developer verify that all mixed strings in the move operation contain valid mixed data.

GEN0109 **FIRST MAP must be map %1, not map %2, for application %3.**

Explanation: The initial map expected by this application is not the map identified in the message. This error occurs when the application starts.

User Response: Record what function you were using before the error occurred, and have the application developer correct the arguments used to start application %3.

GEN0111 **Length of FIRST MAP %1 is not valid.**

Explanation: The length of FIRST MAP %1 received by an application is not the length defined for the map in application %2.

User Response: Have the application developer generate both the application receiving the FIRST MAP and the application issuing the XFER with MAP with the same map definition for the map passed on the XFER.

GEN0119 **Applications %1 and %2 are not compatible.**

Explanation: An application started by a DXFR or CALL is not compatible with the initial application in the transaction or job because the application was generated for a different environment.

User Response: Have the application developer regenerate one or both applications so that the target environments for the applications are the same.

GEN0127 A requested function is not supported for map %1, map group %2.

Explanation: An application requested a map function that is not supported for the specified map and map group. The map group was modified after it was generated and before the application was generated. Some functions that were included in the map group (for example, help maps or EZEMSG on maps) when the application was generated were not in the map group when the map group was generated.

User Response: Have the application developer generate the map group and the application again.

GEN0137 SQL error occurred in work database operation

Explanation: An error occurred during use of the work database when it was implemented using SQL. This message is accompanied by additional SQL diagnostic messages, including GEN0073, that provide additional information about the error.

The run unit ends. Messages are logged.

User Response: Determine the cause of the problem from the SQL code and the SQL error information in related message GEN0074, and correct the database definition.

GEN0184 Application %1 and mapping services program %2 are not compatible.

Explanation: The specified application and mapping services program are generated for different systems.

User Response: Have the application developer generate the mapping services program for the same environment as the application.

GEN0185 Length of %1 for record %2 is not valid and conversion ended.

Explanation: Conversion of a variable length record between the workstation format and host

format cannot be performed because of one of the following conditions:

- - The record is longer than the maximum length defined for the record.
- - The record data ends in the middle of a numeric (BIN, NUM, NUMC, PACK, or PACF) field.
- - The record data ends in the middle of a DBCS character.
- - The record data ends in the middle of a SO/SI string.

User Response: Have the application developer modify the application to set the record length so that it ends on a valid field boundary.

GEN0186 A mixed string in a conversion operation is not valid

Explanation: Conversion of a mixed field from EBCDIC to ASCII or from ASCII to EBCDIC cannot be performed because the double-byte data value is not valid.

User Response: Have the application developer modify the application to make sure that the records to be converted contain valid mixed data.

GEN0187 Conversion table %1 does not support DBCS character conversion.

Explanation: Conversion of a mixed or DBCS field from ASCII to EBCDIC or EBCDIC to ASCII cannot be performed because the specified conversion table does not include conversion tables for double-byte characters.

User Response: Have the application developer modify the application to specify a conversion table that contains the double-byte conversion tables valid for the DBCS or mixed data being converted.

GEN0188 Conversion Error. Function: %1, Return Code: %2, Table: %3

Explanation: A system function was called to perform code page conversion for data used in a client/server program. The function failed.

Possible causes for the failure are:

- The code pages identified in the conversion table are not supported by the conversion functions on your system.
- For DBCS conversion where the source data is in ASCII format, the source data was created under a different DBCS code page than the code page that is currently in effect on the system.

User Response: Correct the cause of the error.

GEN0191 **Application %1, generation date %2, time %3.**

Explanation: An error in application %1 has occurred. Diagnostic messages in the job log explain the nature of the error. Changes to individually generated components of the application may have caused the error.

User Response: Have the application developer verify the generation date and time of the application with that of other generated components.

GEN0192 **Mapping services program %1, generation date %2, time %3.**

Explanation: An error in mapping services program %1 has occurred. Diagnostic messages in the job log explain the nature of the error. Changes to individually generated components of the application may have caused the error.

User Response: Have the application developer verify the generation date and time of the mapping services program with that of other generated components in the application.

GEN0195 **Map format member %1, generation date %2, time %3.**

Explanation: An error in map format member %1 has occurred. Diagnostic messages in the job log explain the nature of the error. Changes to individually generated components of the application may have caused the error.

User Response: Have the application developer verify the generation date and time of the map

format member with that of other generated components in the application.

GEN0210 **IBM VisualAge Generator Server number %1 is not valid.**

Explanation: An attempt was made to start a VisualAge Generator Server routine that does not exist or that is not valid.

User Response: Have the application developer generate and compile the application again to ensure the generated COBOL code has not been modified. Afterward, run the refreshed application. If the problem persists, have the system administrator do all of the following:

- 1 - Record the service number from this message.
 - 2 - Print the job log.
 - 3 - Record the scenario under which this message occurs.
 - 4 - Obtain the COBOL listing and source for the failing application.
 - 5 - Use your electronic link with IBM Service (for example, IBMLINK) if one is available, or contact the IBM Support Center.
-

GEN0232 **Map %1 in map group %2 is not defined or is not supported.**

Explanation: The specified map does not exist or is not defined for the type of device being used.

User Response: Have the application developer either define the map for your device type or select the device for the map. Generate the map group again.

GEN0233 **%1 error on file %2, EZERT8 = %3.**

Explanation: An I/O operation failed for the specified file. This message specifies the COBOL verb performed and the VisualAge Generator file name associated with the operation.

EZERT8 contains either the COBOL/400 status key value or VisualAge Generator file return code, depending on application generation

options. Generation option /SYSCODES causes EZERT8 to contain COBOL/400 status key values; /NOSYSCODES causes EZERT8 to contain VisualAge Generator file return codes.

User Response: Use the appropriate COBOL/400 publication or the VisualAge Generator Return codes publication to diagnose the error, and take the recommended corrective action.

GEN0260 **%1 bytes of UI record won't fit in %2 byte buffer.**

Explanation: The program issued a CONVERSE or XFER with a user interface record. There was not enough room in the communications buffer for the record. The buffer needs space for the record plus any message information written using function EZEUIERR

User Response: Modify the program to reduce the size of the user interface record or write fewer or smaller error messages.

GEN0261 **EZEUIERR message information and inserts won't fit in %1 byte buffer.**

Explanation: The program issued one or more calls to the special function EZEUIERR to write messages associated with a user interface record. The information associated with the last message written won't fit in the buffer used by the program for communicating with user.

User Response: Modify the program to write fewer or smaller error messages.

GEN0262 **Web transaction program and user interface record bean %1 are incompatible.**

Explanation: A Web transaction program was started with information from a user interface record bean that isn't known to the Web program or whose definition is not compatible with the user interface record definition with which the program was generated.

User Response: Insure that the specified bean is defined as the "first ui record" for the program.

Regenerate the program and the Java Beans from the same user interface record definition.

GEN0263 **Number of occurs value %1 is out of range for record array at offset %2**

Explanation: A Web transaction program could not write a user interface record because the "number of occurrences" value set by the program for an array was less than 0 or greater than the maximum number of occurrences defined for the array.

User Response: Correct the program logic so that it sets the number of occurrences to a value within the allowed range.

GEN0264 **Input data entered by the user doesn't fit in user interface record.**

Explanation: A Web transaction program received input data from the Web server that doesn't fit in the interface record. The transaction program and the Java Bean associated with the interface record may have been generated at different times with incompatible user interface record definitions.

User Response: Regenerate the program and the Java Beans from the same definitions. Contact your IBM representative if this doesn't correct the problem.

GEN0265 **Segmented converse is not supported within current function stack.**

Explanation: The program issued a CONVERSE with EZESEGM set to 1 (segmented converse) and at least one of the functions in the current function stack uses parameters or local data items or records. The generated program is not able to save parameters or local storage data over a segmented CONVERSE.

User Response: Modify the program so that the CONVERSE is not used within a function that has parameters or local data.

GEN0266 **MQ function %1, Completion Code %2, Reason Code %3.**

Explanation: The MQ function did not complete successfully, as indicated by the following completion codes:

- 1 MQCC_WARNING
- 2 MQCC_FAILED

The reason for the completion code is set in the reason code field by MQSeries. Some common reason codes are:

- 2009 (Connection broken)
- 2042 (Object already open with conflicting options)
- 2045 (Options not valid for object type)
- 2046 (Options not valid or not consistent)
- 2058 (Queue manager name not valid or not known)
- 2059 (Queue manager not available for connection)
- 2085 (Unknown object name)
- 2086 (Unknown object queue manager)
- 2087 (Unknown remote queue manager)
- 2152 (Object name not valid)
- 2153 (Object queue-manager name not valid)
- 2161 (Queue manager quiescing)
- 2162 (Queue manager shutting down)
- 2201 (Not authorized for access)
- 2203 (Connection shutting down)

User Response: Please refer to the MQSeries Application Programming Reference for further information on MQSeries completion and reason codes.

GEN0267 **Queue Manager Name %1.**

Explanation: This is the name of the queue manager associated with the failing MQ function call listed in message GEN0266.

If the failing MQ function was MQOPEN, MQCLOSE, MQGET, or MQPUT, the name identifies the name identifies the queue manager specified with the object name when the queue was opened. Otherwise, the name is the name of

the queue manager to which the program is connected (or trying to connect).

If the queue manager name is blank, the queue manager is the default queue manager for your system.

User Response: Please refer to the MQSeries Application Programming Reference for further information on the MQSeries completion and reason code listed message GEN0266.

GEN0268 **Queue Name %1.**

Explanation: This is the name of the queue object associated with the failing MQ function call listed in message GEN0266.

User Response: Please refer to the MQSeries Application Programming Reference for further information on MQSeries completion and reason codes reported in message GEN0266.

GEN2001 **The table %1 is not valid for application %2**

Explanation: The reason code is %3. The explanations follow:

- 1 - The table version is not compatible with the current level of IBM VisualAge Generator Server and the running application.
- 2 - The table was generated for an ASCII-based VisualAge Generator runtime environment.
- 3 - The table data is corrupted.
- 4 - The table could not be opened.

User Response: Have the application developer replace table %1 with a correctly generated version.

If the reason code indicates that the table data is corrupted, ensure that the table was transmitted to the host system as a binary image file.

If the reason code indicates the table was generated for an ASCII-based host system, ensure that the table is regenerated for the same target system as the application attempting to use it.

If the reason code indicates the table could not

be opened see previous messages in the job log.

GEN2002 VisualAge Generator Server does not support DBCS data type.

Explanation: VisualAge Generator Server does not support DBCS data type because COBOL/400 does not support DBCS.

User Response: Have the application developer change VisualAge Generator DBCS data types to MIX data types and regenerate the application.

GEN2004 Character conversion from CCSID %1 to %2 is not supported

Explanation: Character conversion is not supported between the two Coded Character Set IDs (CCSID) %1 and %2.

User Response: Have the application developer verify that the specified Coded Character Sets IDs (CCSID) are valid and that conversion between the two CCSIDS is supported. The VisualAge Generator application may have to be regenerated.

GEN2005 Error %1 occurred when converting record %2.

Explanation: EZECONV encountered error code %1 during the call.

User Response: Have the application developer verify that the application logics record %2 with data that matches its definition. The VisualAge Generator application then needs to be regenerated.

GEN2006 The map group %1 is not valid for application %2.

Explanation: The reason code is %3. The explanations follow:

- 1 - Reserved.
- 2 - Reserved.
- 3 - The map group data is corrupted.
- 4 - The map group could not be opened.

User Response: Have the application developer

replace map group %1 with a correctly generated version.

If the reason code indicates that the map group data is corrupted, ensure that the map group was transmitted to the host system as a binary image file.

If the reason code indicates the map group could not be opened see previous messages in the job log.

GEN2007 Press Enter to continue.

Explanation:

User Response:

GEN7025 Error encountered allocating memory.

Explanation: An error was encountered while allocating memory. The system has run out of memory.

User Response: Make sure that you have enough memory on your system as specified in the Software/Hardware requirements for the product. Stop the execution of some of your other applications on your system.

If the problem persists,

- 1. Record the message number and the message text.
- Note: The error message includes the information on where the error occurred and the type of internal error
- 2. Record the situation in which this message occurs.
- 3. Refer to the VisualAge Generator Installation Guide for further instructions on how to report possible defects to the IBM Support Center.

GEN7030 The format of the data descriptor is incorrect. The hex value of the data descriptor in error is %1.

Explanation: The format of the data descriptor is incorrect. A header descriptor is found within the data descriptor.

User Response: Do the following:

- 1. Record the message number and the message text.
- Note: The error message includes the information on where the error occurred and the type of internal error
- 2. Record the situation in which this message occurs.
- 3. Refer to the VisualAge Generator Installation Guide for further instructions on how to report possible defects to the IBM Support Center.

GEN7035 The format of the data descriptor is incorrect.

Explanation: The format of the data descriptor is incorrect. An End Of Description descriptor is not found.

User Response: Do the following:

- 1. Record the message number and the message text.
- Note: The error message includes the information on where the error occurred and the type of internal error
- 2. Record the situation in which this message occurs.
- 3. Refer to the VisualAge Generator Installation Guide for further instructions on how to report possible defects to the IBM Support Center.

GEN7040 The format of the data descriptor is incorrect. An unknown data code %1 was found.

Explanation: The format of the data descriptor is incorrect. An unknown data code was found in the data description.

User Response: Do the following:

- 1. Record the message number and the message text.
- Note: The error message includes the information on where the error occurred and the type of internal error

- 2. Record the situation in which this message occurs.
- 3. Refer to the VisualAge Generator Installation Guide for further instructions on how to report possible defects to the IBM Support Center.

GEN7055 The Conversion Descriptor structure is not valid.

Explanation: The Conversion Descriptor structure CMCVOD required by the conversion routine is incorrect.

User Response: Do the following:

- 1. Record the message number and the message text.
- Note: The error message includes the information on where the error occurred and the type of internal error
- 2. Record the situation in which this message occurs.
- 3. Refer to the VisualAge Generator Installation Guide for further instructions on how to report possible defects to the IBM Support Center.

GEN7065 The data descriptor for parameter %1 is not valid.

Explanation: The data descriptor for the parameter is not valid.

User Response: Do the following:

- 1. Record the message number and the message text.
- Note: The error message includes the information on where the error occurred and the type of internal error
- 2. Record the situation in which this message occurs.
- 3. Refer to the VisualAge Generator Installation Guide for further instructions on how to report possible defects to the IBM Support Center.

GEN9003 VisualAge Generator Server encountered a critical internal processing error.

Explanation: A critical internal processing error was detected. This may include such things as corrupted run unit control blocks, an unexpected return code from an internal function, or illogical code path entry.

Diagnostic messages preceding this message in the job log explain the nature of the error. In most cases, the application or system programmer will need to adjust your application system to correct the problem.

User Response: Either print the job log or record the messages along with the following:

- - The from program name.
- - The to program name.
- - The instruction numbers.

You can view or print the job log with the DSPJOBLOG command. If no diagnostic messages precede this message, ensure that your job logged all messages by checking the 'Message Logging' or LOG value of your job definition or job description, depending on whether the job is interactive or batch. For interactive jobs, command DSPJOB OPTION(*DFNA) will display the 'Message Logging' value.

Contact your application or system programmer with the information you gathered.

GEN9004 VisualAge Generator Server COBOL error handler was invoked to end the run unit.

Explanation: A function check has caused the run unit to end. A database rollback has been issued and heap storage released.

Diagnostic messages preceding this message in the job log explain the nature of the error. In most cases, the application or system programmer will need to adjust your application system to correct the problem.

User Response: Either print the job log or

record the messages along with the following:

- - The from program name.
- - The to program name.
- - The instruction numbers.

You can view or print the job log with the DSPJOBLOG command. If no diagnostic messages precede this message, ensure that your job logged all messages by checking the 'Message Logging' or LOG value of your job definition or job description, depending on whether the job is interactive or batch. For interactive jobs, command DSPJOB OPTION(*DFNA) will display the 'Message Logging' value.

Contact your application or system programmer with the information you gathered.

Chapter 9. Diagnosing problems during run time

This chapter contains information to aid in diagnosing problems that you might encounter while running your VisualAge Generator applications.

VisualAge Generator applications are implemented on OS/400 just like other third-generation language (3GL) applications or programs. This is important to remember, should you or the IBM Support Center need to collect extra ordinary problem diagnostic information in the course of investigating a run-time error.

OS/400 standard diagnostic commands are available to you when diagnosing problems with a VisualAge Generator application. These include such commands as the following:

- ADDTRC (Add Trace Statement)
- STRJOBTRC (Start Job Trace)
- ENDJOBTRC (End Job Trace)
- PRTJOBTRC (Print Job Trace Data)
- STRDBG (Start Debug)
- ENDDBG (End Debug)

For more information on debugging and tracing COBOL programs, refer to the *ILE COBOL/400 Programmer's Guide*.

In most error diagnostic cases, you need to have the following information available when you contact the IBM Support Center:

- The run-time job log, which recorded all messages including second-level text.

To ensure that second-level text is included, change the job before starting the failed scenario using the command `CHGJOB LOG(4 00 *SECLVL)`. When the job ends, the job log is spooled to the assigned output queue. Usually, the most important information in the job log is the escape messages that initiates the abnormal condition, which caused the VisualAge Generator application to end. The program sending the message, the program receiving the message, and the instructions being sent to the program are the key pieces of diagnostic information. The other messages are also important. Be sure to inspect and report the entire job log of information.

- The ILE COBOL/400 compiler listing, which includes the following:
 - VisualAge Generator annotated statements (use generation option `/COMMENTLEVEL=STATEMENTS`)

- VisualAge Generator code production audit comments (use generation option /LINEINFO)
- ILE COBOL/400 source statements (use compiler option OPTION(*SOURCE))
- Any additional spooled files that might have been created as a result of the job ending abnormally, such as dumps or display job snapshots. You can locate all spooled file output from a job by using the WRKJOB command to work with the job. Then you can select the option from the job menu to display spooled output.

Debugging VisualAge Generator

Most logic errors show up when you use the test facility before you generate your application.

During generation, the VisualAge Generator preprocessor checks your application for any remaining syntax errors. In addition, VisualAge Generator also checks that your use of language elements is consistent with the resource association information you selected for each file.

When you run your generated application, different types of errors are detected by VisualAge Generator Server for AS/400, COBOL/400, or OS/400. The error handling varies depending on which product detects the error.

The VisualAge Generator application run-time environment is highly integrated with OS/400 standard debug functions and services. Therefore, VisualAge Generator does not include any additional debug and trace functions. The following VisualAge Generator debug and trace generation options are ignored when generating applications for the OS/400 target environment.

- /DEBUGTRACE
- /ERRDEST
- /LOG
- /TRACE

OS/400 First Failure Data Capture component

VisualAge Generator applications are linked into the OS/400 First Failure Data Capture component when function checks (abends) occur via the OS/400 System Programming Interface (SPI) called Report Software Error (QpdReportSoftwareError). VisualAge Generator uses this function to provide run unit data dumps that can be sent to the IBM Support Center.

VisualAge Generator also uses the report software error SPI function when illogical conditions are detected during run time that might lead to a function

check (abend). In these cases, a unique signature associated with the error condition is provided to the system service. The system service can use the unique signature to scan a service database for the same signature and possible PTFs that can be applied. This expedites the process of problem resolution in some cases.

When job log messages indicate that the OS/400 system problem log has been updated, use the OS/400 command DSPPRB (Display Problem) to view a list of the most recent problems captured on the system; then select the option to display the problem associated with QVGNHS. QVGNHS is the service program that constitutes the majority of VisualAge Generator Server for AS/400. A procedure of this service program issues the SPI function to record the problem. Upon displaying the problem, the menu selections and function keys enable you to display further information, such as spooled files, problem history files, and APAR libraries. The spooled files contain dump data and a copy of the job attributes at the time of the dump.

For more information on using the system problem log in an automated manner, refer to the SystemView System Manager/400 product documentation.

Index

A

about this document xi
activation groups, using with run units 25
application COBOL/400 program, objects generated 13
applications
 moving to another OS/400 system 33
 preparing 17
 VisualAge Generator, running under OS/400 23

B

backup copies of production libraries, maintaining 34

C

changing DDS member types 28
CL, running the preparation 18
COBOL modules, making available 21
coexistence between VisualAge Generator and CSP/AE applications 29
commitment control
 during application run time 28
 ending the cycle 28
 starting the cycle 28
configuring, VisualAge Generator Server for AS/400 1
considerations
 during application run time 28
 VisualAge Generator Server for AS/400 27
customizing
 preparation templates 4
 VisualAge Generator 9
 VisualAge Generator for multiple users 2

D

data definition specifications (DDS)
 changing member types 28
 defined 13
 keywords 27
 modifying 27
 using 27
database files, sharing 30
DDS member types, changing 28

debugging VisualAge Generator 54
declaring parameters and passing arguments 23
detecting errors, VisualAge Generator Server for AS/400 54
diagnosing problems, VisualAge Generator Server for AS/400 53
diagnostic and informational messages 40

E

EFK24PBC, preparation template 5
EFK24PBM, preparation template 7
EFK24PCL, preparation template 5
EFK24PEC, preparation template 5
EFK24PEJ, preparation template 7
EFK24PMN, preparation template 5
EFK24PPM, preparation template 5
EFK24PSC, preparation template 5
EFK24PSM, preparation template 6
EFK24TAM, preparation template 6
EFK24TCM, preparation template 6
EFK24TEM, preparation template 6
EFK24TMJ, preparation template 7
EFK24WCL, preparation template 6
EFK24WSC, preparation template 6
ending, commitment control cycle 28
error detection 54
escape messages 39

F

file and library setup, OS/400 2
files, generation output 11
First Failure Data Capture 54
FIRSTMAP argument, description 24

G

generated applications, preparing 17
generated modules, making available 21
generation
 messages 35
 output files, general 11
 output files, VisualAge Generator Server for AS/400 11

H

how this document is organized xi

I

informational and diagnostic messages 40

J

job description, setting up a 3

K

keywords, data definition specifications 27

L

library and file setup, OS/400 2
library list for a job, establishing 23
listing file, generated objects 14
logical file restrictions 28

M

maintaining backup copies of production libraries 34
map group and table isolation 30
map group format module 14
message file
 CL 15
 compatibility 30
messages
 diagnostic and informational 40
 generation 35
modifying
 data definition specifications 27
 templates 7
moving
 applications to other OS/400 systems 33
 prepared applications to another OS/400 system 33

O

objects generated
 application COBOL/400 program 13
 data definition specification 13
 from generation 11
 listing file 14
 map group format module 14
 message file CL 15
 parts control file 14
 preparation CL 15
 preparation command file 14
 run-time CL 13

- objects generated (*continued*)
 - table binary image file 14
- objects generated for
 - all member types 14
 - application and message table
 - members 15
 - applications 13
 - map groups 14
 - tables 13
- OS/400
 - library and file setup 2
 - publications xv
- output of generation files 11

P

- parts control file 14
- passing arguments and declaring parameters 23
- preparation
 - CL 15
 - CL, running 18
 - command file 14
 - job stream file, described 15

preparation template

- EFK24PBC 5
- EFK24PBJ 7
- EFK24PCL 5
- EFK24PEC 5
- EFK24PEJ 7
- EFK24PMN 5
- EFK24PPM 5
- EFK24PSC 5, 6
- EFK24PSM 6
- EFK24TAM 6
- EFK24TCM 6
- EFK24TEM 6
- EFK24TMJ 7
- EFK24WCL 6
- overview 5

PREPARE subcommand 18

- prepared applications, moving to another OS/400 system 33

preparing

- generated applications 17
- running the preparation CL 18

- problem diagnosing
 - during run time 53

Q

- QVGNHS program 55

R

- reasons for modifying templates 8
- restrictions on logical files 28

- run-time CL
 - objects generated for applications 13

- run-time problems
 - diagnosing 53

run units

- concept in VisualAge Generator 25
- mixing with run unit isolation 29
- using activation groups 25
- using tables 26

running

- generated applications in OS/400 21
- preparation CL 18

S

- setting up an OS/400 job
 - description 3

- sharing native database files 30

- starting, commitment control cycle 28

T

- table binary image file 14

tables

- objects generated for 13
- using with run units 26

templates

- customizing preparation 4
- modifying 7
- reasons for modifying 8

- trademarks vii

U

using

- activation groups with run units 25
- data definition specifications 27
- tables with run units 26

V

VisualAge Generator

- customizing 9

VisualAge Generator Server for AS/400

- diagnosing problems 53
- installing and configuring 1
- messages 39
- modules, making available 21
- overview of generation output files 11

Readers' Comments — We'd Like to Hear from You

VisualAge Generator
Server Guide for AS/400
Version 4.5

Publication No. SH23-0280-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

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

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



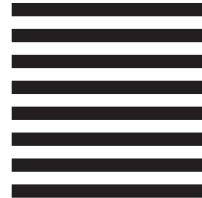
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Information Development
Department G71A / Bldg 062
P.O. Box 12195
Research Triangle Park, NC
27709-2195



Fold and Tape

Please do not staple

Fold and Tape



Program Number: 5622-580
5622-776
5716-VG1
5763-VG1



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

SH23-0280-00

