

IBM Language Environment for VSE/ESA

*Run-Time Migration Guide*  
*Version 1 Release 4*



**Note!**

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

**First Edition (December 1996)**

This edition applies to Version 1 Release 4 of IBM Language Environment for VSE/ESA , 5686-094, and to any subsequent releases until otherwise indicated in new editions or technical newsletters. Make sure you are using the correct edition for the level of the product.

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

A form for reader's comments is provided at the back of this publication. If the form has been removed, address your comments to:

IBM Corporation  
Attn: Dept. ECJ - BP/003D  
6300 Diagonal Highway  
Boulder, CO 80301,  
U.S.A.

or to

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

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

© **Copyright International Business Machines Corporation 1995, 1996.**

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

---

# Contents

- Tables..... V**
  
- Notices..... vii**
  - Programming Interface Information.....vii
  - Trademarks.....vii
  
- About This Book.....ix**
  - What Is LE/VSE?..... ix
  - LE/VSE-Conforming Languages..... ix
  - LE/VSE Compatibility with Previous Versions of COBOL..... x
  - Terms Used in This Book.....x
  
- Using Your Documentation..... xi**
  
- Summary Of Changes..... xiii**
  - Major Changes to the Product..... xiii
  - Release 4, December 1996..... xiii
  
- Chapter 1. Planning to Migrate to LE/VSE..... 1**
  - Checklist for Migration..... 1
  - Planning to Link and Run with LE/VSE.....3
  
- Chapter 2. Migrating from a Previous LE/VSE Release..... 5**
  - Run-Time Options Considerations.....5
  - New Run-Time Options..... 5
  - LE/370-Compatibility Run-Time Options Removed..... 5
  - TEST|NOTEST Run-Time Option.....5
  - RPTOPTS Report Differences..... 6
  - RPTSTG Report Differences..... 6
  - Installation Default Differences..... 6
  - Abnormal Termination Considerations.....6
  
- Chapter 3. Migrating from Other Run-Time Environments..... 9**
  - Compatibility with Previous Run-Time Libraries..... 9
  - Migrating C/370 Applications to LE/VSE..... 9
  - Migrating DOS/VS COBOL and VS COBOL II Applications to LE/VSE..... 10
  - Migrating DOS PL/I Applications to LE/VSE..... 10
  - Migrating ILC Applications to LE/VSE.....10
  - Migrating Assembler Applications to LE/VSE.....11
  
- Chapter 4. Choosing Run-Time Options.....13**
  - How LE/VSE Run-Time Options Differ from C/370, DOS/VS COBOL, VS COBOL II, and DOS PL/I Options..... 13
  - C/370 and LE/VSE Run-Time Options Comparison..... 13
  - DOS/VS COBOL and LE/VSE Run-Time Options Comparison.....14
  - VS COBOL II and LE/VSE Run-Time Options Comparison..... 15
  - DOS PL/I and LE/VSE Run-Time Options Comparison..... 17
  - Recommended Settings for C, COBOL, and PL/I Applications..... 18
  - Setting LE/VSE Options for C Applications..... 19

Setting LE/VSE Options for COBOL Applications.....	21
Setting LE/VSE Options for PL/I Applications.....	24
<b>Chapter 5. Other HLL Migration Considerations.....</b>	<b>29</b>
C Considerations.....	29
Standard Streams.....	29
Passing Command Line Parameters.....	29
Prefix of strerror() Messages in C.....	29
Storage Report.....	29
COBOL Considerations.....	29
Abends.....	29
ALL31 Run-Time Option and AMODE.....	30
Run-Time Message Output.....	30
Storage Report.....	30
PL/I Considerations.....	30
Dumps.....	30
Condition Handling.....	30
Run-Time Message Output.....	31
Format and Content of Messages in PL/I.....	31
Storage Report.....	31
<b>Chapter 6. Compatibility with Other Products.....</b>	<b>33</b>
Machine Requirements.....	33
Programming Requirements.....	33
Required Licensed Programs.....	33
Optional Licensed Programs.....	33
Vendor Products That Support LE/VSE.....	34
<b>Bibliography.....</b>	<b>35</b>
Language Environment Publications .....	35
LE/VSE-Conforming Language Product Publications.....	35
Softcopy Publications.....	36

---

# Tables

1. LE/VSE-Conforming Languages.....	x
2. How to Use LE/VSE and Language Publications.....	xi
3. Comparison of Run-Time Option Installation Default Values (Batch).....	6
4. Comparison of Run-Time Option Installation Default Values (CICS).....	6
5. C/370 Migration Considerations.....	9
6. COBOL Migration Considerations.....	10
7. DOS PL/I Migration Considerations.....	10
8. ILC Migration Considerations.....	10
9. C/370 and LE/VSE Options.....	13
10. DOS/VS COBOL and LE/VSE Options.....	14
11. VS COBOL II and LE/VSE Options.....	15
12. DOS PL/I and LE/VSE Options.....	17
13. Setting LE/VSE Options for C Applications.....	19
14. Setting LE/VSE Options for COBOL Applications.....	21
15. Setting LE/VSE Options for PL/I Applications.....	24
16. Required Licensed Programs for LE/VSE.....	33
17. Optional Licensed Compiler Programs for LE/VSE.....	33
18. Optional Licensed Programs for LE/VSE.....	34



## 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:

IBM Director of Licensing  
IBM Corporation  
500 Columbus Avenue  
Thornwood, NY 10594  
U.S.A.

## Programming Interface Information

---

This book is intended to help with application programming. This book documents General-Use Programming Interface and Associated Guidance Information provided by IBM Language Environment for VSE/ESA.

General-Use programming interfaces allow the customer to write programs that obtain the services of IBM Language Environment for VSE/ESA.

## Trademarks

---

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

BookManager	IBM	QMF
C/370	Integrated Language	S/390
CICS	Environment	SQL/DS
CICS/VSE	Language Environment	VSE/ESA
DFSORT	MVS	
	OS/400	





## About This Book

---

This book describes the steps you must take to run applications with IBM Language Environment for VSE/ESA. The book is intended for application developers who are migrating to IBM Language Environment for VSE/ESA, but not necessarily to a new language compiler. Familiarity with the run-time libraries of the different languages, and understanding of the basics of linking and running applications, are assumed.

Information in *Run-Time Migration Guide* will not provide a comprehensive guide to the migration process, but rather, will help you create a broad migration strategy. This book will help you identify which modules can be migrated first, and which will require relinking or recompiling. It also explains how to use LE/VSE run-time options to achieve behavior compatible with your old modules. Refer to one or more of the following manuals for such details as how to upgrade old source code into new code:

- [IBM C for VSE/ESA Migration Guide](#)
- [IBM COBOL for VSE/ESA Migration Guide](#)
- [IBM PL/I for VSE/ESA Migration Guide](#)

## What Is LE/VSE?

---

LE/VSE is a set of common services and language-specific routines that provide a single run-time environment for applications written in *LE/VSE-conforming* versions of the C, COBOL, and PL/I high level languages (HLLs), and for many applications written in previous versions of COBOL. (For a list of LE/VSE-conforming languages, and a description of compatibility with previous versions of COBOL, see [“LE/VSE Compatibility with Previous Versions of COBOL”](#) on page x.) LE/VSE also supports applications written in assembler language using LE/VSE-provided macros and assembled using High Level Assembler (HLASM).

Prior to LE/VSE, each programming language provided its own separate run-time environment. LE/VSE combines essential and commonly-used run-time services—such as message handling, condition handling, storage management, date and time services, and math functions—and makes them available through a set of interfaces that are consistent across programming languages. With LE/VSE, you can use one run-time environment for your applications, regardless of the application's programming language or system resource needs, because most system dependencies have been removed.

Services that work with only one language are available within language-specific portions of LE/VSE.

LE/VSE consists of:

- Basic routines for starting and stopping programs, allocating storage, communicating with programs written in different languages, and indicating and handling error conditions.
- Common library services, such as math services and date and time services, that are commonly needed by programs running on the system. These functions are supported through a library of callable services.
- Language-specific portions of the common run-time library.

LE/VSE is the implementation of Language Environment on the VSE platform. Language Environment is offered on two other platforms: on MVS and VM as IBM Language Environment for MVS & VM, and on OS/400 as Integrated Language Environment.

## LE/VSE-Conforming Languages

---

An LE/VSE-conforming language is any HLL that adheres to the LE/VSE common interface. [Table 1 on page x](#) lists the LE/VSE-conforming language compiler products you can use to generate applications that run with LE/VSE Release 4.

Table 1. LE/VSE-Conforming Languages

Language	LE/VSE-Conforming Language	Minimum Release
C	IBM C for VSE/ESA	Release 1
COBOL	IBM COBOL for VSE/ESA	Release 1
PL/I	IBM PL/I for VSE/ESA	Release 1

Any HLL not listed in Table 1 on page x is known as a *non-LE/VSE-conforming* or, alternatively, a *pre-LE/VSE-conforming* language. Some examples of non-LE/VSE-conforming languages are: C/370, DOS/VS COBOL , VS COBOL II, and DOS PL/I.

Only the following products can generate applications that run with LE/VSE :

- LE/VSE-conforming languages
- HLASM using LE/VSE-provided macros (for details, see [LE/VSE Programming Guide](#)) )
- DOS/VS COBOL and VS COBOL II, with some restrictions (see [“LE/VSE Compatibility with Previous Versions of COBOL”](#) on page x below)

## LE/VSE Compatibility with Previous Versions of COBOL

Although DOS/VS COBOL and VS COBOL II are non-LE/VSE-conforming languages, many applications generated with these compilers can run with LE/VSE without recompiling or relink-editing. For details about compatibility, see [Chapter 3, “Migrating from Other Run-Time Environments,”](#) on page 9. .

VS COBOL II can also dynamically call some LE/VSE date and time callable services. For details, see [LE/VSE Programming Reference](#) .

## Terms Used in This Book

Unless otherwise stated, the following terms are used in this book to refer to the specified languages:

### Term...

**Refers to the language supported by...**

### C

The IBM C for VSE/ESA compiler

### COBOL

The IBM COBOL for VSE/ESA and VS COBOL II compilers

### PL/I

The IBM PL/I for VSE/ESA compilers

For a list of LE/VSE-conforming language compilers, see [“LE/VSE-Conforming Languages”](#) on page ix.

# Using Your Documentation

The publications provided with LE/VSE are designed to help you:

- Manage the run-time environment for applications written in LE/VSE -conforming languages.
- Write applications that use the LE/VSE callable services.
- Develop interlanguage communication (ILC) applications.
- Plan for, install, customize, and maintain LE/VSE.
- Debug problems in your LE/VSE-conforming applications.
- Migrate your high-level language applications to LE/VSE.

Language programming information is provided in the high-level language programming manuals that provide language definition, library function syntax and semantics, and programming guidance information.

Each publication helps you perform a different task. For a complete list of publications you might need, see [“Bibliography” on page 35](#).

*Table 2. How to Use LE/VSE and Language Publications*

<b>To...</b>	<b>Use...</b>	
Evaluate LE/VSE	<a href="#">LE/VSE Fact Sheet</a>	GC33-6679
	<a href="#">LE/VSE Concepts Guide</a>	GC33-6680
Plan for, install, customize, and maintain LE/VSE	<a href="#">LE/VSE Customization Guide</a>	SC33-6682
Understand the LE/VSE program models and concepts	<a href="#">LE/VSE Concepts Guide</a>	GC33-6680
	<a href="#">LE/VSE Programming Guide</a>	SC33-6684
Find syntax for LE/VSE run-time options and callable services	<a href="#">LE/VSE Programming Reference</a>	SC33-6685
Develop your LE/VSE-conforming applications	<a href="#">LE/VSE Programming Guide</a>	SC33-6684
	<i>Your language programming guide</i>	
	<a href="#">LE/VSE C Run-Time Programming Guide</a>	SC33-6688
	<a href="#">LE/VSE C Run-Time Library Reference</a>	SC33-6689
Develop interlanguage communication (ILC) applications	<a href="#">LE/VSE Writing Interlanguage Communication Applications</a>	SC33-6686
Debug your LE/VSE-conforming application and get details on run-time messages	<a href="#">LE/VSE Debugging Guide and Run-Time Messages</a>	SC33-6681
Migrate applications to LE/VSE	<i>Run-Time Migration Guide</i>	SC33-6687
	<i>Your language migration guide</i>	
Diagnose problems that occur in your LE/VSE-conforming application	<a href="#">LE/VSE Debugging Guide and Run-Time Messages</a>	SC33-6681
Understand warranty information	<a href="#">LE/VSE Licensed Program Specifications</a>	GC33-6683



# Summary Of Changes

---

This section lists the major changes that have been made to LE/VSE since Release 1.

## Major Changes to the Product

---

### Release 4, December 1996

LE/VSE Release 4 is a major functional enhancement of LE/VSE Release 1 (program number 5686-067); there were no intermediate releases. This release number was chosen to match IBM Language Environment for MVS & VM Release 4, upon which LE/VSE Release 4 is based.

- C language run-time support has been added for C applications compiled with an LE/VSE-conforming C language compiler.
- Support has been added for interactive and batch-mode debugging of applications using a debug tool such as Debug Tool for VSE/ESA. The TEST run-time option specifies the conditions under which a debug tool assumes control when the user application is being initialized.
- The following run-time options have been added:

#### **ARGPARSE**

Specifies whether arguments on the command line are to be parsed in the usual C format.

#### **ENV**

Specifies the operating environment for a C application.

#### **ENVAR**

Sets the initial values for the environment variables specified. With ENVAR, you can pass into an application switches or tagged information that can then be accessed during application execution using the C functions `getenv()`, `setenv()`, and `clearenv()`.

#### **EXECOPS**

Specifies whether run-time options can be specified on the command line.

#### **PLIST**

Specifies the format of the invocation parameters received by your C application when it is invoked.

#### **REDIR**

Specifies whether re-directions for `stdin`, `stderr`, and `stdout` are allowed from the command line.

#### **TRACE**

Determines whether LE/VSE run-time library tracing is active.

- The CEE5CIB callable service has been added. CEE5CIB returns a pointer to a condition information block (CIB) associated with a given condition token. CEE5CIB is used only during condition handling.
- The CEECBLDY callable service has been added. CEECBLDY converts a string representing a date into a COBOL Integer format that is compatible with ANSI COBOL intrinsic functions.
- LE/VSE locale callable services that access and manage locale information have been added:

#### **CEEFMON**

Formats monetary strings. CEEFMON corresponds to the C function `strfmon()`.

#### **CEEFTDS**

Formats time and date into a character string. CEEFTDS corresponds to the C function `strftime()`.

#### **CEELCNV**

Queries locale numeric conventions. CEELCNV corresponds to the C function `localeconv()`.

**CEEQDTC**

Queries locale date and time conventions and returns the specified format information. CEEQDTC corresponds to the C function `localdtconv()`.

**CEEQRYL**

Queries the active locale environment. CEEQRYL corresponds to the C function `setlocale()`.

**CEESCOL**

Compares the collation weight of two strings. CEESCOL corresponds to the C function `strcoll()`.

**CEESETL**

Sets the locale operating environment. CEESETL corresponds to the C function `setlocale()`.

**CEESTXF**

Transforms string characters into collation weights. CEESTXF corresponds to the C function `strxfrm()`.

- Predefined locales for specifying different national language and cultural conventions have been added.
- Locale definition utility for modifying and creating locales has been added (requires LE/VSE-conforming C language compiler).
- Nested enclaves can now be created by the C `system()` function.

For more information about changes between LE/VSE Release 1 and LE/VSE Release 4, see [Chapter 2, “Migrating from a Previous LE/VSE Release,”](#) on page 5. .

---

# Chapter 1. Planning to Migrate to LE/VSE

This chapter provides a migration checklist to help you plan the migration of your applications to the LE/VSE run time. It also contains information on planning to link and run your applications.

---

## Checklist for Migration

Each task in the checklist is necessary and you should perform them in the order shown.

### 1. Educate yourself and other programmers about LE/VSE.

Ensure that you and other application programmers who will be involved in the migration effort are familiar with the features of LE/VSE and the differences between your current run-time environment and LE/VSE. You can get information about LE/VSE directly from LE/VSE publications and from user groups such as SHARE and GUIDE. You will discover many of the differences between run-time environments as you complete the other steps in this checklist.

### 2. Take an inventory of the applications and vendor products you intend to run with LE/VSE.

CCOBOL, and PL/I programs

- For each program you intend to move to the LE/VSE run time, gather the following information:
  - Version and release of the compiler that generated the program
  - Which COBOL programs were compiled with RES and which with NORES
  - Run-time options used and how they were specified
  - Which programs call or are called by assembler programs
  - Which applications contain ILC
  - Which programs are used with CICS, DL/I, or SQL/DS
  - Control statements used
  - Frequency and types of abends
  - Test cases required and test cases available
  - Amount of storage used
  - Frequency of execution of reusable or common modules
  - Program execution time (both CPU and elapsed)

Vendor tools, packages, and products

- Ensure that all vendor tools, packages, and products run with LE/VSE, and that any source code for packages is compatible with your LE/VSE -conforming compiler.
- Ensure that any vendor code generators generate code that is compatible with your LE/VSE-conforming compiler.
- Ensure that vendor development tools and debuggers will not issue their own STXIT PC or STXIT AB macros. In the event of a program check or abend, the LE/VSE condition manager must get control first.

For information on how to obtain a list of vendor products that work with LE/VSE, see [“Vendor Products That Support LE/VSE”](#) on page 34.

### 3. Prioritize programs.

Determine the effort required to migrate each program and the order in which you will migrate them. Each program will require some level of effort to migrate, ranging from minimal testing to a code rewrite. Using the information from your inventory analysis, determine whether each program:

- Requires minimum, moderate, or extensive testing

- Runs with LE/VSE without change
- Requires relinking with LE/VSE
- Requires recompiling with LE/VSE-conforming compiler, without change to the source code
- Requires source code change
- Does not run with LE/VSE

After you have determined the effort required to migrate each program, list them in the order you want to move them to LE/VSE, taking into account the importance of each program and the frequency of execution.

#### 4. Plan for and install LE/VSE.

Perform the following tasks, which can be done concurrently:

- Assess storage requirements.

Both DASD and virtual storage requirements might be larger for LE/VSE than for your current run-time environment:

- During conversion, you might need DASD storage for the LE/VSE run time as well as your current run-time library. For LE/VSE DASD storage requirements, see [LE/VSE Customization Guide](#).
- Virtual storage requirements for placing library routines above and below the 16MB line might increase. For more information about the virtual storage required for library routines, see [LE/VSE Customization Guide](#).
- Virtual storage requirements for allocating run-time storage above or below the 16MB line might also increase. The amount of increase depends on which LE/VSE storage options you specify. For a list of recommended settings, see [“Recommended Settings for C, COBOL, and PL/I Applications” on page 18](#).

- Install the required components of LE/VSE.

LE/VSE is made up of four run-time components: LE/VSE base library routines, C library routines, COBOL library routines, and PL/I library routines. You can install either all of the LE/VSE components or install the LE/VSE base and one or more of the language-specific components. The latter allows you to migrate your applications one language at a time, with applications written in one language running with LE/VSE while your other applications run with their current run-time libraries.

Migrate applications that contain ILC only after you have migrated your C -, COBOL-, and PL/I-only applications. (An application that contains assembler, but is otherwise created from one language, is not considered an ILC application in this book.)

- Change default run-time options as appropriate.

To ensure that the LE/VSE run-time results are compatible with your current run-time results, you will need to change some of the default settings for the run-time options. For a list of recommended settings, see [“Recommended Settings for C, COBOL, and PL/I Applications” on page 18](#).

- Determine whether to add LE/VSE to the permanent LIBDEF search chains for each required partition, or to the temporary LIBDEF search chain for the partition when you run your job.

Once you add LE/VSE to the permanent LIBDEF search chains for a partition, it is available to all applications that run in that partition. Ensure all applications are functioning correctly with LE/VSE before adding LE/VSE to the permanent LIBDEF search chains. You might consider adding LE/VSE to a partition's temporary LIBDEF search chains until you have confirmed this.

You can choose to phase in the LE/VSE run time gradually by using the temporary LIBDEF search chain approach, in which case you phase in one CICS or batch partition at a time. Although using this approach means changing JCL, a gradual conversion can be easier than moving all of your applications at one time.

#### 5. Set up a regression testing procedure.

To ensure that the LE/VSE run-time results are compatible with your current run-time results, you will need to perform regression tests on all the programs you migrate. Run your applications in parallel



with both your current run time and with the LE/VSE run time to confirm that the results are the same. You can temporarily add LE/VSE to the LIBDEF search chains to accomplish this. Once your applications are running with LE/VSE in a test environment, take performance measurements, especially on any time-critical or response-critical applications.

#### 6. Move applications into production.

When your testing shows the entire application receives the expected results, you can move the entire unit over to production use. However, in case of unexpected errors, be prepared for instant recovery:

- Under CICS, return to the last commit point and then continue processing from that point using the unmigrated program.
- For batch applications, use your backup and restore facilities to recover.

After you move your applications to production use with the LE/VSE run time, monitor your applications for a while to ensure that they continue to work properly. You can then run with the confidence that you had in your previous run time.

## Planning to Link and Run with LE/VSE

---

If LE/VSE is installed in the default sublibraries, the sublibrary PRD2.SCEEBASE contains resident routines that are linked with the application and are used to resolve external references at link-edit time. LE/VSE callable services and other routines, such as those for initialization and termination, are also located in PRD2.SCEEBASE.

You will need to modify the LIBDEF job control statements in your input stream to point to PRD2.SCEEBASE.

See [LE/VSE Programming Guide](#) for detailed information on linking and running your applications.



---

## Chapter 2. Migrating from a Previous LE/VSE Release

LE/VSE Release 4 provides general object and phase compatibility for applications that run with a previous release of LE/VSE.

Phases will run compatibly with any level of LE/VSE that is equivalent to or higher than the level used to link-edit them.

Object modules can be link-edited with any level of LE/VSE that is equivalent to or higher than the level required by the compiler that generated them.

---

### Run-Time Options Considerations

The following sections describe how LE/VSE Release 4 run-time options differ from LE/VSE Release 1 run-time options.

#### New Run-Time Options

The following run-time options are new with LE/VSE Release 4:

- ARGPARSE
- ENV
- ENVAR
- EXECOPS
- PLIST
- REDIR
- TRACE

For information about these run-time options, see [LE/VSE Programming Reference](#).

#### LE/370-Compatibility Run-Time Options Removed

With LE/VSE Release 1, you could specify the following run-time options for compatibility with LE/370:

- CBLQDA
- FLOW|NOFLOW
- INTERRUPT
- SIMVRD|NOSIMVRD
- VCTRSAVE

If you specified any of these run-time options, it was syntax-checked, but had no effect on the run-time behavior of your application. If the syntax of the option was correct, you did not receive a run-time message. With LE/VSE Release 4, these run-time options are no longer supported for compatibility. If you specify any of these run-time options, you receive run-time message CEE3609I, and the option is ignored. For information about run-time messages, see [LE/VSE Debugging Guide and Run-Time Messages](#).

#### TEST|NOTEST Run-Time Option

With LE/VSE Release 1, you could also specify the TEST|NOTEST run-time option for compatibility with LE/370. As with other LE/370-compatibility run-time options, if you specified the TEST|NOTEST run-time option, it was syntax-checked but had no effect on the run-time behavior of your application. With LE/VSE Release 4, the TEST|NOTEST run-time option is fully supported. For more information about the TEST|NOTEST run-time option, see [LE/VSE Programming Reference](#).

## RPTOPTS Report Differences

With LE/VSE Release 1, the run-time options report produced by the RPTOPTS(ON) run-time option included the LE/370-compatibility run-time options listed in “LE/370-Compatibility Run-Time Options Removed” on page 5 and the PL/I-compatibility run-time options ISAINC and ISASIZE. With LE/VSE Release 4, these run-time options are not included in the report. For more information about the report produced by the RPTOPTS(ON) run-time option, see [LE/VSE Programming Reference](#).

## RPTSTG Report Differences

With LE/VSE Release 1, the storage report produced by the RPTSTG(ON) run-time option used the phrases "Successful GETMAINS issued" and "Successful FREEMAINS issued" to describe the number of requestsLE/VSE issued to the operating system (or CICS) to get storage and to release storage. With LE/VSE Release 4, these phrases are replaced by "Number of segments allocated" and "Number of segments freed", respectively. For more information about the report produced by the RPTSTG(ON) run-time option, see [LE/VSE Programming Reference](#).

## Installation Default Differences

Several run-time options have different IBM-supplied installation default values in LE/VSE Release 4 than in LE/VSE Release 1. The following table lists those run-time options that have different installation default values in batch.

*Table 3. Comparison of Run-Time Option Installation Default Values (Batch)*

LE/VSE Release 1 Default	LE/VSE Release 4 Default
ANYHEAP(32K,16K,ANYWHERE,FREE)	ANYHEAP(16K,8K,ANYWHERE,FREE)
BELOWHEAP(32K,16K,FREE)	BELOWHEAP(8K,4K,FREE)
HEAP(64K,64K,ANYWHERE,KEEP,16K,16K)	HEAP(32K,32K,ANYWHERE,KEEP,8K,4K)
LIBSTACK(32K,16K,FREE)	LIBSTACK(8K,4K,FREE)
NOTEST(NONE,*,NOPROMPT,*)	NOTEST(ALL,*,PROMPT,")
STACK(512K,512K,BELOW,KEEP)	STACK(128K,128K,BELOW,KEEP)

The following table lists those run-time options that have different installation default values in CICS.

*Table 4. Comparison of Run-Time Option Installation Default Values (CICS)*

LE/VSE Release 1 Default	LE/VSE Release 4 Default
NOTEST(NONE,*,NOPROMPT,*)	NOTEST(ALL,*,PROMPT,")
TERMTHDACT(TRACE)	TERMTHDACT(MSG)

For information about customizing run-time option installation default values, see [LE/VSE Customization Guide](#).

## Abnormal Termination Considerations

With LE/VSE Release 1, the IBM-supplied assembler user exit for CICS always requested an abend when an enclave ended with the following types of unhandled LE/VSE condition of severity 2 or greater, regardless of the setting of the ABTERMENC run-time option:

- A software-raised condition, such as the condition raised by LE/VSE if you try to run an AMODE 24 program without specifying the ALL31(OFF) and STACK(,BELOW) run-time options
- A user-raised condition (raised by a call to the CEESGL callable service)

When the assembler user exit requests an abend at enclave termination, LE/VSE uses an abend code provided by the exit or, if the exit does not provide an abend code, one based upon the severity of the condition that caused termination. The IBM-supplied assembler user exit for CICS does not provide an abend code, so LE/VSE uses an abend code based upon the condition severity. A severity 2 condition produces an abend code of 2000, a severity 3 condition produces an abend code of 3000, and a severity 4 condition produces an abend code of 4000.

With LE/VSE Release 4, the assembler user exit for CICS does not specifically request an abend when an enclave terminates with an unhandled condition of severity 2 or greater. Instead, the ABTERMENC run-time option in effect at the time is honored. If ABTERMENC(ABEND) is in effect during abnormal termination, the enclave is terminated with abend code 4038.

In addition, with LE/VSE Release 4, the IBM-supplied abnormal termination exit for CICS, which is driven whenever an enclave terminates abnormally, requests a CICS transaction dump with a dump code of 4039. The CICS transaction dump is produced in addition to the abnormal termination information produced by LE/VSE under the control of the TERMTHDACT run-time option. Unlike LE/VSE abnormal termination information, which is written to the CESE transient data queue, the CICS dump is written to the CICS dump dataset. There was no corresponding abnormal termination exit for CICS supplied with LE/VSE Release 1.

For more information about customizing the assembler user exit and abnormal termination exit for CICS, see [LE/VSE Customization Guide](#).



---

# Chapter 3. Migrating from Other Run-Time Environments

This chapter describes, in general, the compatibility of LE/VSE with previous run-time libraries. It also describes what you must do to migrate different object modules and phases to LE/VSE.

Not all migration considerations are described in this book. For a more detailed description of language-specific migration considerations, see the appropriate language migration guide:

- [IBM C for VSE/ESA Migration Guide](#)
- [IBM COBOL for VSE/ESA Migration Guide](#)
- [IBM PL/I for VSE/ESA Migration Guide](#)

---

## Compatibility with Previous Run-Time Libraries

LE/VSE provides object compatibility with certain previous COBOL run-time libraries, as follows:

### DOS/VS COBOL and VS COBOL II

With certain exceptions, such as those listed in [Table 6 on page 10](#) and [Table 8 on page 10](#), LE/VSE provides object compatibility for applications generated with the following pre-LE/VSE IBM language products:

- DOS/VS COBOL Release 3.1
- VS COBOL II Release 3.2 or later

Subject to these exceptions, phases created with these compilers, and link-edited with their associated run-time libraries, will run compatibly with LE/VSE without relinking.

Also subject to the exceptions, object modules created with the above compilers can be link-edited and run with LE/VSE without recompiling.

LE/VSE does not provide object compatibility with the following run-time libraries:

### C/370

LE/VSE does not provide object compatibility for applications generated with the C/370 compiler. For migration information, see [Table 5 on page 9](#).

### DOS PL/I

LE/VSE does not provide object compatibility for applications generated with the DOS PL/I compiler. For migration information, see [Table 7 on page 10](#).

---

## Migrating C/370 Applications to LE/VSE

---

*Table 5. C/370 Migration Considerations*

---

To Migrate:	You Need To:
A phase containing one or more C/370 programs	<ol style="list-style-type: none"><li>1. Compile the C source code with C/VSE. If the compile is unsuccessful, upgrade the source code to ensure it follows the ANSI standard, and recompile.</li><li>2. Link-edit the phase with LE/VSE.</li></ol>

---

## Migrating DOS/VS COBOL and VS COBOL II Applications to LE/VSE

A subset of DOS/VS COBOL and VS COBOL II compatibility exceptions is listed below.

*Table 6. COBOL Migration Considerations*

<b>To Migrate:</b>	<b>You Need To:</b>
A phase containing one or more DOS/VS COBOL programs or VS COBOL II NORES programs, that is dynamically called by a VS COBOL II or COBOL/VSE program	Relink the phase with LE/VSE.
A phase containing one or more DOS/VS COBOL or VS COBOL II programs with calls to C/370 or DOS PL/I	See <a href="#">Table 8 on page 10</a> for instructions.
A phase containing one or more VS COBOL II NORES programs that require the facilities provided by LE/VSE run-time options	Relink the phase with LE/VSE.
A phase containing one or more DOS/VS COBOL programs that require the facilities provided by LE/VSE run-time options	<ol style="list-style-type: none"><li>1. Upgrade the DOS/VS COBOL source, as necessary, and compile with COBOL/VSE.</li><li>2. Link-edit the phase with LE/VSE.</li></ol>

## Migrating DOS PL/I Applications to LE/VSE

*Table 7. DOS PL/I Migration Considerations*

<b>To Migrate:</b>	<b>You Need To:</b>
A phase containing one or more DOS PL/I programs	<ol style="list-style-type: none"><li>1. Upgrade the DOS PL/I source, as necessary, and compile with PL/I VSE.</li><li>2. Link-edit the phase with LE/VSE.</li></ol>

## Migrating ILC Applications to LE/VSE

*Table 8. ILC Migration Considerations*

<b>To Migrate:</b>	<b>You Need To:</b>
A phase containing one or more DOS/VS COBOL programs, with calls to or from DOS PL/I	<ol style="list-style-type: none"><li>1. Upgrade the DOS/VS COBOL source code, as necessary, and compile with COBOL/VSE .</li><li>2. Upgrade the DOS PL/I source code, as necessary, and compile with PL/I VSE .</li><li>3. Link-edit the phase with LE/VSE.</li></ol>
A phase containing one or more VS COBOL II programs, with calls to or from DOS PL/I	<ol style="list-style-type: none"><li>1. Upgrade the DOS PL/I source code, as necessary, and compile with PL/I VSE .</li><li>2. Relink the phase with LE/VSE.</li></ol>



---

Table 8. ILC Migration Considerations (continued)

---

<b>To Migrate:</b>	<b>You Need To:</b>
A phase containing one or more DOS/VS COBOL programs, with calls to or from C/370	<ol style="list-style-type: none"><li data-bbox="862 247 1446 310">1. Upgrade the DOS/VS COBOL source code, as necessary, and compile with COBOL/VSE .</li><li data-bbox="862 321 1446 384">2. Upgrade the C/370 source code, as necessary, and compile with C/VSE .</li><li data-bbox="862 394 1446 426">3. Link-edit the phase with LE/VSE.</li></ol>
A phase containing one or more VS COBOL II programs, with calls to or from C/370	<ol style="list-style-type: none"><li data-bbox="862 468 1446 531">1. Upgrade the C/370 source code, as necessary, and compile with C/VSE .</li><li data-bbox="862 541 1446 562">2. Link-edit the phase with LE/VSE.</li></ol>

---

## Migrating Assembler Applications to LE/VSE

---

To run assembler applications with LE/VSE, you need to code the applications with the assembler macros provided by LE/VSE. You must also ensure the assembler programs adhere to certain conventions for register and storage usage. See the chapter on assembler considerations in [LE/VSE Programming Guide](#).



---

## Chapter 4. Choosing Run-Time Options

This chapter provides information on how LE/VSE run-time options differ from HLL-specific run-time options. It also provides recommended settings for C, COBOL, and PL/I applications running in both CICS and non-CICS environments.

### How LE/VSE Run-Time Options Differ from C/370, DOS/VS COBOL, VS COBOL II, and DOS PL/I Options

---

The following tables show how LE/VSE run-time options differ from C/370 -, DOS/VS COBOL-, VS COBOL II-, and DOS PL/I-specific run-time options. If you don't find a specific HLL run-time option in the tables, you can assume they operate under LE/VSE in the same way they did before LE/VSE.

#### C/370 and LE/VSE Run-Time Options Comparison

Table 9. C/370 and LE/VSE Options

C/370 Option	LE/VSE Equivalent	Notes
ISAINC ( <i>incr_size</i> )	STACK ( <i>incr_size</i> )	The C/370 ISAINC run-time option is mapped to the LE/VSE STACK run-time option for compatibility. It affects all languages in the enclave. If you don't change the C/370 ISAINC option, you will receive an informational message at run time.
ISASIZE ( <i>init_size</i> )	STACK ( <i>init_size</i> )	The C/370 ISASIZE run-time option is mapped to the LE/VSE STACK run-time option for compatibility. It affects all languages in the enclave. If you don't change the C/370 ISASIZE option, you will receive an informational message at run time.
LANGUAGE	NATLANG	The C/370 LANGUAGE run-time option is mapped to the LE/VSE NATLANG run-time option for compatibility. It affects all languages in the enclave. If you don't change the C/370 LANGUAGE option, you will receive an informational message at run time.
REPORT	RPTSTG(ON)	The C/370 REPORT run-time option is mapped to the LE/VSE RPTSTG(ON) run-time option for compatibility. It affects all languages in the enclave. If you don't change the C/370 REPORT option, you will receive an informational message at run time.
NOREPORT	RPTSTG(OFF)	The C/370 NOREPORT run-time option is mapped to the LE/VSE RPTSTG(OFF) run-time option for compatibility. It affects all languages in the enclave. If you don't change the C/370 NOREPORT option, you will receive an informational message at run time.

Table 9. C/370 and LE/VSE Options (continued)

<b>C/370 Option</b>	<b>LE/VSE Equivalent</b>	<b>Notes</b>
SPIE	TRAP(ON)	The C/370 SPIE run-time option is mapped to the LE/VSE TRAP(ON) run-time option for compatibility. It affects all languages in the enclave. The mapping of SPIE might differ depending upon other options specified. For more information, see <a href="#">LE/VSE Programming Reference</a> . If you don't change the C/370 SPIE option, you will receive an informational message at run time.
NOSPIE	TRAP(OFF)	The C/370 NOSPIE run-time option is mapped to the LE/VSE TRAP(OFF) run-time option for compatibility. It affects all languages in the enclave. The mapping of NOSPIE might differ depending upon other options specified. For more information, see <a href="#">LE/VSE Programming Reference</a> . If you don't change the C/370 NOSPIE option, you will receive an informational message at run time.
STAE	TRAP(ON)	The C/370 STAE run-time option is mapped to the LE/VSE TRAP(ON) run-time option for compatibility. It affects all languages in the enclave. The mapping of STAE might differ depending upon other options specified. For more information, see <a href="#">LE/VSE Programming Reference</a> . If you don't change the C/370 STAE option, you will receive an informational message at run time.
NOSTAE	TRAP(OFF)	The C/370 NOSTAE run-time option is mapped to the LE/VSE TRAP(OFF) run-time option for compatibility. It affects all languages in the enclave. The mapping of NOSTAE might differ depending upon other options specified. For more information, see <a href="#">LE/VSE Programming Reference</a> . If you don't change the C/370 NOSTAE option, you will receive an informational message at run time.

## DOS/VS COBOL and LE/VSE Run-Time Options Comparison

Table 10. DOS/VS COBOL and LE/VSE Options

<b>DOS/VS COBOL Option</b>	<b>LE/VSE Equivalent</b>	<b>Notes</b>
A (SYSPARM)	AIXBLD	The LE/VSE AIXBLD run-time option is compatible with the DOS/VS COBOL SYSPARM='A' run-time option. It affects only COBOL programs in the enclave.
NA (SYSPARM)	NOAIXBLD	The LE/VSE NOAIXBLD run-time option is compatible with the DOS/VS COBOL SYSPARM='NA' run-time option. It affects only COBOL programs in the enclave.
D (SYSPARM)	DEBUG	The LE/VSE DEBUG run-time option is compatible with the DOS/VS COBOL SYSPARM='D' run-time option. It affects only COBOL programs in the enclave.
ND (SYSPARM)	NODEBUG	The LE/VSE NODEBUG run-time option is compatible with the DOS/VS COBOL SYSPARM='ND' run-time option. It affects only COBOL programs in the enclave.

Table 10. DOS/VS COBOL and LE/VSE Options (continued)

DOS/VS COBOL Option	LE/VSE Equivalent	Notes
UPSI	UPSI	The LE/VSE UPSI run-time option replaces the DOS/VS COBOL UPSI run-time option provided by the // UPSI job control statement. The UPSI switches set by the // UPSI job control statement are not available to COBOL programs under LE/VSE.

## VS COBOL II and LE/VSE Run-Time Options Comparison

Table 11. VS COBOL II and LE/VSE Options

VS COBOL II Option	LE/VSE Equivalent	Notes
AIXBLD	AIXBLD	The LE/VSE AIXBLD run-time option is compatible with the VS COBOL II AIXBLD run-time option. It affects only COBOL programs in the enclave.
NOAIXBLD	NOAIXBLD	The LE/VSE NOAIXBLD run-time option is compatible with the VS COBOL II NOAIXBLD run-time option. It affects only COBOL programs in the enclave.
DEBUG	DEBUG	The LE/VSE DEBUG run-time option is compatible with the VS COBOL II DEBUG run-time option. It affects only COBOL programs in the enclave.
NODEBUG	NODEBUG	The LE/VSE NODEBUG run-time option is compatible with the VS COBOL II NODEBUG run-time option. It affects only COBOL programs in the enclave.
LANGUAGE	NATLANG	The VS COBOL II LANGUAGE run-time option is mapped to the LE/VSE NATLANG run-time option for compatibility. It affects all languages in the enclave. If you don't change the VS COBOL II LANGUAGE option, you will receive an informational message at run time.
LIBKEEP	Not applicable	There is no LE/VSE equivalent for the VS COBOL II LIBKEEP run-time option. To obtain similar performance function, use the Library Routine Retention (LRR) feature described in <a href="#">LE/VSE Programming Guide</a> . If you don't remove the VS COBOL II LIBKEEP option, you will receive an informational message at run time.
NOLIBKEEP	Not applicable	There is no LE/VSE equivalent for the VS COBOL II NOLIBKEEP run-time option. If you don't remove the VS COBOL II NOLIBKEEP option, you will receive an informational message at run time.
MIXRES	Not applicable	There is no LE/VSE equivalent for the VS COBOL II MIXRES run-time option. MIXRES applications supported by LE/VSE always exhibit RES behavior. For more information, see <a href="#">IBM COBOL for VSE/ESA Migration Guide</a> . If you don't remove the VS COBOL II MIXRES option, you will receive an informational message at run time.

Table 11. VS COBOL II and LE/VSE Options (continued)

<b>VS COBOL II Option</b>	<b>LE/VSE Equivalent</b>	<b>Notes</b>
NOMIXRES	Not applicable	There is no LE/VSE equivalent for the VS COBOL II NOMIXRES run-time option. MIXRES applications supported by LE/VSE always exhibit RES behavior. For more information, see <i>IBM COBOL for VSE/ESA Migration Guide</i> . If you don't remove the VS COBOL II NOMIXRES option, you will receive an informational message at run time.
RTEREUS	RTEREUS	The LE/VSE RTEREUS run-time option is compatible with the VS COBOL II RTEREUS run-time option. The RTEREUS option is intended for use when the main program of an enclave is a COBOL program. The RTEREUS option can cause problems for HLLs other than COBOL.
NORTEREUS	NORTEREUS	The VS COBOL II NORTEREUS run-time option is compatible with the VS COBOL II NORTEREUS run-time option.
SIMVRD	Not applicable	There is no LE/VSE equivalent for the VS COBOL II SIMVRD run-time option. If you don't remove the VS COBOL II SIMVRD option, you will receive an informational message at run time.
NOSIMVRD	Not applicable	There is no LE/VSE equivalent for the VS COBOL II NOSIMVRD run-time option. If you don't remove the VS COBOL II NOSIMVRD option, you will receive an informational message at run time.
SPOUT	RPTOPTS(ON) RPTSTG(ON)	The VS COBOL II SPOUT run-time option is mapped to the LE/VSE RPTOPTS(ON) and RPTSTG(ON) run-time options for compatibility. It affects all languages in the enclave. If you don't change the VS COBOL II SPOUT option, you will receive an informational message at run time.
NOSPOUT	RPTOPTS(OFF) RPTSTG(OFF)	The VS COBOL II NOSPOUT run-time option is mapped to the LE/VSE RPTOPTS(OFF) and RPTSTG(OFF) run-time options for compatibility. It affects all languages in the enclave. If you don't change the VS COBOL II NOSPOUT option, you will receive an informational message at run time.
SSRANGE	CHECK(ON)	The VS COBOL II SSRANGE run-time option is mapped to the LE/VSE CHECK(ON) run-time option for compatibility. It affects only COBOL programs in the enclave. If you don't change the VS COBOL II SSRANGE option, you will receive an informational message at run time.
NOSSRANGE	CHECK(OFF)	The VS COBOL II NOSSRANGE run-time option is mapped to the LE/VSE CHECK(OFF) run-time option for compatibility. It affects only COBOL programs in the enclave. If you don't change the VS COBOL II NOSSRANGE option, you will receive an informational message at run time.

Table 11. VS COBOL II and LE/VSE Options (continued)

VS COBOL II Option	LE/VSE Equivalent	Notes
STAE	TRAP(ON)	The VS COBOL II STAE run-time option is mapped to the LE/VSE TRAP(ON) run-time option for compatibility. It affects all languages in the enclave. The mapping of STAE might differ depending upon other options specified. For more information, see <a href="#">LE/VSE Programming Reference</a> . If you don't change the VS COBOL II STAE option, you will receive an informational message at run time.
NOSTAE	TRAP(OFF)	The VS COBOL II NOSTAE run-time option is mapped to the LE/VSE TRAP(OFF) run-time option for compatibility. It affects all languages in the enclave. The mapping of NOSTAE might differ depending upon other options specified. For more information, see <a href="#">LE/VSE Programming Reference</a> . If you don't change the VS COBOL II NOSTAE option, you will receive an informational message at run time.
UPSI	UPSI	The VS COBOL II UPSI option is processed for compatibility.
WSCLEAR	STORAGE(00)	The VS COBOL II WSCLEAR run-time option is not supported under LE/VSE. For behavior similar to that produced by the VS COBOL II WSCLEAR run-time option, use the LE/VSE STORAGE(00) run-time option. For more information, see <a href="#">IBM COBOL for VSE/ESA Migration Guide</a> . If you don't change the VS COBOL II WSCLEAR option, you will receive an informational message at run time, and your application might not behave as expected.
NOWSCLEAR	STORAGE(NONE)	The VS COBOL II NOWSCLEAR run-time option is not supported under LE/VSE. For behavior similar to that produced by the VS COBOL II NOWSCLEAR run-time option, use the LE/VSE STORAGE(NONE) run-time option. For more information, see <a href="#">IBM COBOL for VSE/ESA Migration Guide</a> . If you don't change the VS COBOL II NOWSCLEAR option, you will receive an informational message at run time, and your application might not behave as expected.

## DOS PL/I and LE/VSE Run-Time Options Comparison

Table 12. DOS PL/I and LE/VSE Options

DOS PL/I Option	LE/VSE Equivalent	Notes
COUNT	Not applicable	There is no LE/VSE equivalent for the DOS PL/I COUNT run-time option. If you don't remove the DOS PL/I COUNT option, you will receive an informational message at run time.
NOCOUNT	Not applicable	There is no LE/VSE equivalent for the DOS PL/I NOCOUNT run-time option. If you don't remove the DOS PL/I NOCOUNT option, you will receive an informational message at run time.
FLOW	Not applicable	There is no LE/VSE equivalent for the DOS PL/I FLOW run-time option. If you don't remove the DOS PL/I FLOW option, you will receive an informational message at run time.

Table 12. DOS PL/I and LE/VSE Options (continued)

DOS PL/I Option	LE/VSE Equivalent	Notes
NOFLOW	Not applicable	There is no LE/VSE equivalent for the DOS PL/I NOFLOW run-time option. If you don't remove the DOS PL/I NOFLOW option, you will receive an informational message at run time.
ISASIZE ( <i>init_size</i> )	STACK ( <i>init_size</i> )	The DOS PL/I ISASIZE run-time option is mapped to the LE/VSE STACK run-time option for compatibility. It affects all languages in the enclave. If you don't change the DOS PL/I ISASIZE option, you will receive an informational message at run time.
REPORT	RPTSTG(ON)	The DOS PL/I REPORT run-time option is mapped to the LE/VSE RPTSTG(ON) run-time option for compatibility. It affects all languages in the enclave. If you don't change the DOS PL/I REPORT option, you will receive an informational message at run time.
NOREPORT	RPTSTG(OFF)	The DOS PL/I NOREPORT run-time option is mapped to the LE/VSE RPTSTG(OFF) run-time option for compatibility. It affects all languages in the enclave. If you don't change the DOS PL/I NOREPORT option, you will receive an informational message at run time.
STAE	TRAP(ON)	The DOS PL/I STAE run-time option is mapped to the LE/VSE TRAP(ON) run-time option for compatibility. It affects all languages in the enclave. The mapping of STAE might differ depending upon other options specified. For more information, see <a href="#">LE/VSE Programming Reference</a> . If you don't change the DOS PL/I STAE option, you will receive an informational message at run time.
NOSTAE	TRAP(OFF)	The DOS PL/I NOSTAE run-time option is mapped to the LE/VSE TRAP(OFF) run-time option for compatibility. It affects all languages in the enclave. The mapping of NOSTAE might differ depending upon other options specified. For more information, see <a href="#">LE/VSE Programming Reference</a> . If you don't change the DOS PL/I NOSTAE option, you will receive an informational message at run time.

## Recommended Settings for C, COBOL, and PL/I Applications

The following tables show the LE/VSE run-time options defaults and the recommended settings for C, COBOL, and PL/I applications running in both CICS and non-CICS environments. Language-specific run-time options only appear in the table for the specific language. If you don't find an option in the tables, you can assume that the recommended settings for CICS and non-CICS are the same as the LE/VSE default setting. For ILC applications, use the recommendations that best suit the combination of languages used.



## Setting LE/VSE Options for C Applications

Table 13. Setting LE/VSE Options for C Applications

Recommended for Batch	Same as LE/VSE Default for Batch?	Recommended for CICS	Same as LE/VSE Default for CICS?	Notes
ABPERC(NONE)	Yes	Not applicable		
ABTERMENC(RETCODE)	Yes	ABTERMENC(ABEND)	Yes	TRAP(ON) must be in effect for ABTERMENC to have an effect when the unhandled condition is a program check or an abend. ABTERMENC is always in effect for unhandled conditions raised by the CEESGL callable service, regardless of the setting of the TRAP option.
ALL31(ON)	No—See notes	ALL31(ON)	Yes	The LE/VSE default for batch applications is ALL31(OFF).  If your application (batch or CICS) contains only AMODE 31 programs, specify ALL31(ON). If your application contains any AMODE 24 programs, you must specify ALL31(OFF). If you use specify ALL31(OFF), you must also specify STACK(,BELOW), as AMODE 24 programs usually require stack storage below the 16MB line.
ANYHEAP(16K,8K, ANYWHERE,FREE)	Yes	ANYHEAP(4K,4K, ANYWHERE,FREE)	Yes	
ARGPARSE	Yes	Not applicable		
BELOWHEAP(8K,4K,FREE)	Yes	BELOWHEAP(4K,4K,FREE)	Yes	
ENVAR("")	Yes	ENVAR("")	Yes	C applications can access the environment variables using the C function <code>getenv()</code> .
EXECOPS	Yes	Not applicable		
HEAP(32K,32K, ANYWHERE,KEEP,8K,4K)	Yes	HEAP(4K,4K, ANYWHERE,KEEP,4K,4K)	Yes	
LIBSTACK(8K,4K,FREE)	Yes	LIBSTACK(4K,4K,FREE)	Yes	

Table 13. Setting LE/VSE Options for C Applications (continued)

Recommended for Batch	Same as LE/VSE Default for Batch?	Recommended for CICS	Same as LE/VSE Default for CICS?	Notes
MSGFILE(filename) (Any name is acceptable for the output file.)	See note	Not applicable		The LE/VSE default is MSGFILE(SYSLST).
PLIST(HOST)	Yes	Not applicable		
REDIR	Yes	Not applicable		ARGPARSE must be in effect for REDIR to work.
STACK(128K, 128K, ANYWHERE, KEEP)	No—See notes	STACK(4K, 4K, ANYWHERE, KEEP)	Yes	<p>The LE/VSE default for batch applications is STACK(128K, 128K, BELOW, KEEP).</p> <p>If your application (batch or CICS) contains only AMODE 31 programs, specify STACK(, ANYWHERE) and ALL31(ON). Applications running with ALL31(OFF) must specify STACK(, BELOW) to ensure that stack storage is addressable by the application.</p>
STORAGE(NONE, NONE, NONE, 8K)	Yes	STORAGE(NONE, NONE, NONE, 0K)	Yes	
TERMTHDACT(TRACE)	Yes	TERMTHDACT(MSG)	Yes	The contents of the dump produced by TERMTHDACT depend, in part, on the setting of the TRACE run-time option.
TRACE(OFF, 4K, DUMP, LE=0)	Yes	TRACE(OFF, 4K, DUMP, LE=0)	Yes	Under normal termination conditions, if TRACE(ON, DUMP) is in effect, a dump containing only a trace table is produced, regardless of the TERMTHDACT settings. Under abnormal termination conditions, however, the contents of the dump depend on the settings of the TERMTHDACT and TRACE run-time options.

Table 13. Setting LE/VSE Options for C Applications (continued)

Recommended for Batch	Same as LE/VSE Default for Batch?	Recommended for CICS	Same as LE/VSE Default for CICS?	Notes
TRAP(ON)	Yes	TRAP(ON)	Yes	TRAP(ON) must be in effect for applications to run successfully. The TRAP option is similar to the SPIE   NOSPIE and STAE   NOSTAE options used by C/370. In an options string with TRAP(ON) or TRAP(OFF) together with SPIE   NOSPIE and/or STAE   NOSTAE, the TRAP setting takes precedence over all others.

## Setting LE/VSE Options for COBOL Applications

Table 14. Setting LE/VSE Options for COBOL Applications

Recommended for Batch	Same as LE/VSE Default for Batch?	Recommended for CICS	Same as LE/VSE Default for CICS?	Notes
ABPERC(NONE)	Yes	Not applicable		
ABTERMENC(ABEND)	No—See notes	ABTERMENC(ABEND)	Yes	The LE/VSE default for batch applications is ABTERMENC(RETCODE). For behavior compatible with pre-LE/VSE COBOL, specify ABTERMENC(ABEND).  TRAP(ON) must be in effect for ABTERMENC to have an effect when the unhandled condition is a program check or an abend. ABTERMENC is always in effect for unhandled conditions raised by the CEESGL callable service, regardless of the setting of the TRAP option.

Table 14. Setting LE/VSE Options for COBOL Applications (continued)

Recommended for Batch	Same as LE/VSE Default for Batch?	Recommended for CICS	Same as LE/VSE Default for CICS?	Notes
NOAIXBLD	Yes	Not applicable		Access method services (AMS) messages are directed to the MSGFILE <i>filename</i> or, if the file identified by <i>filename</i> is unavailable, to SYSLST.
ALL31(ON)	No—See notes	ALL31(ON)	Yes	The LE/VSE default for batch applications is ALL31(OFF).  If your application (batch or CICS) contains only AMODE 31 programs, specify ALL31(ON). If your application contains a VS COBOL II NORES program, a DOS/VS COBOL program, or any AMODE 24 programs, you must specify ALL31(OFF). If you specify ALL31(OFF), you must also specify STACK(,BELOW), as AMODE 24 programs usually require stack storage below the 16MB line.
ANYHEAP(16K,8K, ANYWHERE,FREE)	Yes	ANYHEAP(4K,4K, ANYWHERE,FREE)	Yes	
BELOWHEAP(8K,4K,FREE)	Yes	BELOWHEAP(4K,4K,FREE)	Yes	
CBLOPTS(ON)	Yes	Not applicable		
Not applicable		CBLPSHPOP(ON)	Yes	
HEAP(32K,32K, ANYWHERE,KEEP,8K,4K)	Yes	HEAP(4K,4K, ANYWHERE,KEEP,4K,4K)	Yes	
LIBSTACK(8K,4K,FREE)	Yes	LIBSTACK(4K,4K,FREE)	Yes	
MSGFILE(filename) (Any name is acceptable for the output file.)	See note	Not applicable		The LE/VSE default is MSGFILE(SYSLST).
NORTEREUS	Yes	Not applicable		

Table 14. Setting LE/VSE Options for COBOL Applications (continued)

Recommended for Batch	Same as LE/VSE Default for Batch?	Recommended for CICS	Same as LE/VSE Default for CICS?	Notes
STACK(64K,64K, ANYWHERE,KEEP)	No—See notes	STACK(4K,4K, ANYWHERE,KEEP)	Yes	<p>The LE/VSE default for batch applications is STACK(128K,128K,BELOW,KEEP).</p> <p>If your application (batch or CICS) contains only AMODE 31 programs, specify STACK(,ANYWHERE) and ALL31(ON). Applications running with ALL31(OFF) must specify STACK(,BELOW) to ensure that stack storage is addressable by the application.</p>
STORAGE(NONE, NONE,NONE,8K)	Yes	STORAGE(NONE, NONE,NONE,0K)	Yes	<p>For behavior similar to that produced by the VS COBOL II programs running with the WSCLEAR run-time option, or DOS/VS COBOL programs running under CICS, specify STORAGE(00,NONE,NONE,8K).</p>
TERMTHDACT(DUMP)	No—See notes	TERMTHDACT(MSG)	Yes	<p>The LE/VSE default for batch applications is TERMTHDACT(TRACE). For behavior compatible with pre-LE/VSE COBOL, specify TERMTHDACT(DUMP).</p> <p>The contents of the dump produced by TERMTHDACT depend, in part, on the setting of the TRACE run-time option.</p>

Table 14. Setting LE/VSE Options for COBOL Applications (continued)

Recommended for Batch	Same as LE/VSE Default for Batch?	Recommended for CICS	Same as LE/VSE Default for CICS?	Notes
TRACE(OFF,4K, DUMP,LE=0)	Yes	TRACE(OFF,4K, DUMP,LE=0)	Yes	Under normal termination conditions, if TRACE(ON,,DUMP) is in effect, a dump containing only a trace table is produced, regardless of the TERMTHDACT settings. Under abnormal termination conditions, however, the contents of the dump depend on the settings of the TERMTHDACT and TRACE run-time options.
TRAP(ON)	Yes	TRAP(ON)	Yes	TRAP(ON) must be in effect for applications to run successfully. The TRAP option is similar to the STAE   NOSTAE option used by VS COBOL II. In an options string with TRAP(ON) or TRAP(OFF) together with STAE or NOSTAE, the TRAP setting takes precedence over all others.

## Setting LE/VSE Options for PL/I Applications

Table 15. Setting LE/VSE Options for PL/I Applications

Recommended for Batch	Same as LE/VSE Default for Batch?	Recommended for CICS	Same as LE/VSE Default for CICS?	Notes
ABPERC(NONE)	Yes	Not applicable		

Table 15. Setting LE/VSE Options for PL/I Applications (continued)

Recommended for Batch	Same as LE/VSE Default for Batch?	Recommended for CICS	Same as LE/VSE Default for CICS?	Notes
ABTERMENC(ABEND)	No—See notes	ABTERMENC(ABEND)	Yes	The LE/VSE default for batch applications is ABTERMENC(RETCODE). For behavior compatible with DOS PL/I, specify ABTERMENC(ABEND).  TRAP(ON) must be in effect for ABTERMENC to have an effect when the unhandled condition is a program check or an abend. ABTERMENC is always in effect for unhandled conditions raised by the CEESGL callable service, regardless of the setting of the TRAP option.
ALL31(ON)	No—See notes	ALL31(ON)	Yes	The LE/VSE default for batch applications is ALL31(OFF).  If your application (batch or CICS) contains only AMODE 31 programs, specify ALL31(ON). If your application contains any AMODE 24 programs, you must specify ALL31(OFF). If you specify ALL31(OFF), you must also specify STACK(,BELOW), as AMODE 24 programs usually require stack storage below the 16MB line.
ANYHEAP(16K,8K, ANYWHERE,FREE)	Yes	ANYHEAP(4K,4K, ANYWHERE,FREE)	Yes	
BELOWHEAP(8K,4K,FREE)	Yes	BELOWHEAP(4K,4K,FREE)	Yes	
DEPTHCONDLMT(10)	Yes—See note	DEPTHCONDLMT(10)	Yes—See note	For behavior compatible with DOS PL/I in batch or under CICS, specify DEPTHCONDLMT(0).

Table 15. Setting LE/VSE Options for PL/I Applications (continued)

Recommended for Batch	Same as LE/VSE Default for Batch?	Recommended for CICS	Same as LE/VSE Default for CICS?	Notes
ERRCOUNT(0)	No—See notes	ERRCOUNT(0)	No—See notes	The LE/VSE default for batch and CICS applications is ERRCOUNT(20). For correct behavior of PL/I applications, specify ERRCOUNT(0).
HEAP(32K,32K, ANYWHERE,KEEP,8K,4K)	Yes	HEAP(4K,4K, ANYWHERE,KEEP,4K,4K)	Yes	Heap storage is sensitive to the AMODE of the requestor and the main procedure. Storage is only allocated above the 16MB line when all of the following conditions occur: the requestor is running in AMODE 31; HEAP(,,ANYWHERE) is in effect; the main procedure is AMODE 31.
LIBSTACK(8K,4K,FREE)	Yes	LIBSTACK(4K,4K,FREE)	Yes	
MSGFILE(filename) (Any name is acceptable for the output file.)	See note	Not applicable		The LE/VSE default is MSGFILE(SYSLST).
STACK(128K,128K, ANYWHERE,KEEP)	No—See notes	STACK(4K,4K, ANYWHERE,KEEP)	Yes	The LE/VSE default for batch applications is STACK(128K,128K,BELOW,KEEP).  If your application (batch or CICS) contains only AMODE 31 programs, specify STACK(,,ANYWHERE) and ALL31(ON). Applications running with ALL31(OFF) must specify STACK(,,BELOW) to ensure that stack storage is addressable by the application.
STORAGE(NONE, NONE,NONE,8K)	Yes	STORAGE(NONE, NONE,NONE,0K)	Yes	
TERMTHDACT(TRACE)	Yes	TERMTHDACT(MSG)	Yes	The contents of the dump produced by TERMTHDACT depend, in part, on the setting of the TRACE run-time option.



Table 15. Setting LE/VSE Options for PL/I Applications (continued)

<b>Recommended for Batch</b>	<b>Same as LE/VSE Default for Batch?</b>	<b>Recommended for CICS</b>	<b>Same as LE/VSE Default for CICS?</b>	<b>Notes</b>
TRACE(OFF,4K, DUMP,LE=0)	Yes	TRACE(OFF,4K, DUMP,LE=0)	Yes	Under normal termination conditions, if TRACE(ON,,DUMP) is in effect, a dump containing only a trace table is produced, regardless of the TERMTHDACT settings. Under abnormal termination conditions, however, the contents of the dump depend on the settings of the TERMTHDACT and TRACE run-time options.
TRAP(ON)	Yes	TRAP(ON)	Yes	TRAP(ON) must be in effect for applications to run successfully. The TRAP option is similar to the STAE   NOSTAE option used by DOS PL/I. In an options string with TRAP(ON) or TRAP(OFF) together with STAE or NOSTAE, the TRAP setting takes precedence over all others.
XUFLOW(ON   AUTO)	See notes	XUFLOW(ON   AUTO)	See notes	The LE/VSE default for batch and CICS applications is XUFLOW(AUTO). Either XUFLOW(ON) or XUFLOW(AUTO) meets the requirement of PL/I language semantics that exponent underflow be signaled.



---

## Chapter 5. Other HLL Migration Considerations

This chapter discusses a few language-specific migration considerations, including differences in how LE/VSE and an HLL handle return codes and run-time messages. For a complete list of migration considerations, see the appropriate migration guide:

- [IBM C for VSE/ESA Migration Guide](#)
- [IBM COBOL for VSE/ESA Migration Guide](#)
- [IBM PL/I for VSE/ESA Migration Guide](#)

---

### C Considerations

The following sections list some of the C migration considerations.

#### Standard Streams

In C/370, you could override the destination of error messages by redirecting `stderr`. LE/VSE determines the destination of all messages from the `MSGFILE` run-time option. See the section on the `MSGFILE` option in [LE/VSE Programming Guide](#) for further information.

#### Passing Command Line Parameters

In C/370, if an error was detected with the parameters being passed to the main program, the program terminated with a return code of 8 and a message indicating the reason the program terminated. Under LE/VSE, the same message is displayed, but the program also terminates with a 4093 abend, reason code 52 (hexadecimal 34). For further information about reason codes, see [LE/VSE Debugging Guide and Run-Time Messages](#).

#### Prefix of `strerror()` Messages in C

With LE/VSE, all `strerror()` messages in C contain a prefix. With C/370, there was no prefix on these messages. The prefix is `EDCxxxxa` where `xxxx` is a number (always `5xxx`) and the `a` is either I, W, or E. See [LE/VSE Debugging Guide and Run-Time Messages](#) for a list of C messages.

#### Storage Report

The format of the run-time storage report generated by the LE/VSE `RPTSTG` run-time option is different from the format of the storage reports produced by the C/370 `REPORT` run-time option. For more information about the `RPTSTG` run-time option, see [LE/VSE Programming Reference](#).

---

### COBOL Considerations

The following sections list some of the COBOL migration considerations.

#### Abends

In DOS/VS COBOL and VS COBOL II, a severe unhandled error condition always resulted in an abend. With LE/VSE, you use the `ABTERMENC` run-time option to specify whether a severe unhandled condition results in an abend or a normal termination with a return code and reason code. The IBM-supplied installation value for the `ABTERMENC` run-time option is `ABTERMENC(RETCODE)`. To ensure that your application ends with an abend when there is a severe unhandled condition, specify `ABTERMENC(ABEND)`.

## ALL31 Run-Time Option and AMODE

When you link-edit a VS COBOL II (or COBOL/VSE) program compiled with the NORENT compiler option, the default addressing mode of the link-edited phase is AMODE ANY. This might result in your program being invoked in 24-bit addressing mode. In order to specify the ALL31(ON) run-time option, your program must be invoked in 31-bit addressing mode. Therefore, you should link-edit your application as AMODE 31. You can use the MODE linkage editor control statement to override the default addressing mode.

## Run-Time Message Output

LE/VSE directs run-time messages produced by VS COBOL II (and COBOL/VSE) programs to the file specified by the LE/VSE MSGFILE run-time option. Run-time messages produced by DOS/VS COBOL programs, such as output from the DOS/VS COBOL SYMDMP, STATE, FLOW, and COUNT compiler options are directed to SYSLST.

For more information about the MSGFILE run-time option, see [LE/VSE Programming Reference](#). For more information about run-time message output from COBOL programs, see [LE/VSE Programming Guide](#).

LE/VSE manages all user-specified output directed to the system-logical output device. This includes output produced by the following statements:

- DISPLAY [UPON SYSLST]
- EXHIBIT (DOS/VS COBOL only)
- READY TRACE (DOS/VS COBOL only)

**Note:** The COBOL DISPLAY statement is not supported under CICS. The DOS/VS COBOL READY TRACE and EXHIBIT statements are also not supported under CICS.

For more information about how LE/VSE determines the destination of user-specified output from COBOL programs, see [LE/VSE Programming Reference](#).

## Storage Report

The format of the run-time storage report generated by the LE/VSE RPTSTG run-time option is different from the format of the storage reports produced by the VS COBOL II SPOUT run-time option. For more information about the RPTSTG run-time option, see [LE/VSE Programming Reference](#).

## PL/I Considerations

---

The following sections list some of the PL/I migration considerations.

### Dumps

The output produced by PLIDUMP is different when running under LE/VSE. For detailed information, see [IBM PL/I for VSE/ESA Migration Guide](#)

### Condition Handling

In general, PL/I condition handling continues to function in the same way when running under LE/VSE; however, you should consider the following:

- The ERRCOUNT run-time option specifies how many conditions of severity 2, 3, and 4 can occur before the enclave terminates abnormally. The IBM-supplied installation value for the ERRCOUNT run-time option is ERRCOUNT(20). This value is not suitable for all PL/I applications. To ensure that your application behaves correctly, and is compatible with DOS PL/I behavior, specify ERRCOUNT(0).
- The diagnostic message for an ERROR condition is issued only if there is no ERROR ON-unit established, or if the ERROR ON-unit does not recover from the condition by using a GOTO out of block. However, for other PL/I conditions whose implicit action includes printing a message and raising the ERROR condition, the message is issued before control is given to an established ERROR ON-unit.

## Run-Time Message Output

LE/VSE directs run-time message output from PL/I programs to the file specified by the LE/VSE MSGFILE run-time option, instead of to the PL/I SYSPRINT STREAM PRINT file. User-specified output is still directed to the PL/I SYSPRINT STREAM PRINT file. If you want LE/VSE to handle this output, specify the run-time option MSGFILE(SYSPRINT). When you specify MSGFILE(SYSPRINT), run-time messages and user-specified output are routed to SYSLST. For more information about the MSGFILE run-time option, see [LE/VSE Programming Reference](#).

## Format and Content of Messages in PL/I

The format and content of run-time messages is different for PL/I applications running with LE/VSE. If you have applications that analyze run-time output, you must change them. Differences include the following:

- The message number in the message prefix is now four digits instead of three digits.
- The message severity in the message prefix can now be I, W, E, S, or C.
- The message text of some mixed-case English and Japanese messages has been enhanced.

For more information about using and handling messages, see [LE/VSE Programming Reference](#).

## Storage Report

The format of the run-time storage report generated by the LE/VSE RPTSTG run-time option is different than the format of the storage reports produced by the DOS PL/I REPORT run-time option. For more information about the RPTSTG run-time option, see [LE/VSE Programming Reference](#).



---

## Chapter 6. Compatibility with Other Products

This chapter lists products that are compatible with LE/VSE.

### Machine Requirements

---

LE/VSE-conforming compiler-generated object code runs on any hardware configuration supported by the licensed programs specified below. LE/VSE supports the DBCS character sets on the IBM Personal System/55 (as 3270) and IBM 5550 Family (as 3270).

### Programming Requirements

---

LE/VSE runs under the control of, or in conjunction with, the following IBM licensed programs and their subsequent releases unless otherwise announced by IBM.

#### Required Licensed Programs

The licensed programs listed in [Table 16 on page 33](#) are required to install and customize LE/VSE, or to run LE/VSE applications.

---

*Table 16. Required Licensed Programs for LE/VSE*

Required Licensed Program	Minimum Release	Program Number
One of:		
VSE/ESA	Version 1 Release 4	5750-ACD
VSE/ESA	Version 2 Release 1	5690-VSE
High Level Assembler/MVS & VM & VSE	Release 1	5696-234

#### Optional Licensed Programs

The licensed compiler programs listed in [Table 17 on page 33](#), with or without the Debug Tool feature, can optionally be used to generate LE/VSE applications.

---

*Table 17. Optional Licensed Compiler Programs for LE/VSE*

Optional Licensed Program	Minimum Release	Program Number
C/VSE	Release 1	5686-A01
COBOL/VSE	Release 1	5686-068
PL/I VSE	Release 1	5686-069

The licensed programs listed in [Table 18 on page 34](#) can optionally be used with LE/VSE.

Table 18. Optional Licensed Programs for LE/VSE

Optional Licensed Program	Minimum Release	Program Number
BookManager Read	Release 2 is required to view softcopy documentation	73F6-023 (Read/2)
CICS/VSE <sup>1</sup> on page 34	Version 2 Release 3 with PTF UN89454	5686-026
CSP/AD	Version 3 Release 3	5668-813
CSP/AE	Version 3 Release 3	5668-814
DFSORT/VSE	Version 3 Release 1	5746-SM3
DL/I DOS/VS	Release 10 with PTF UN73450	5746-XX1
DOS/VS Sort/Merge	Version 2 Release 5	5746-SM2
QMF/VSE	Version 3 Release 2	5648-061
REXX/VSE	Release 1	5686-058
SQL/DS	Version 3 Release 4 with PTF UN76254	5688-103

**Note:**

1. LE/VSE is not supported in VSE/ICCF interactive partitions.

## Vendor Products That Support LE/VSE

For a list of vendor tools and application packages that are enabled to work with LE/VSE, ask your IBM representative to obtain *Language Environment Enabled Vendor Tools and Application Packages* from either the IBM Marketing Tools disk (by issuing TOOLCAT MKTTOOLS GET LEVNRST PACKAGE) or the S/390 Developers Association Forum Disk. You can also call 1 800 IBM 3333, ext. 703, to request a copy. This document provides details on each of the vendor products.



# Bibliography

---

## Language Environment Publications

---

[LE/VSE Fact Sheet](#)  
[LE/VSE Concepts Guide](#)  
[LE/VSE Debugging Guide and Run-Time Messages](#)  
[LE/VSE Customization Guide](#)  
[LE/VSE Licensed Program Specifications, GC33-6683](#)  
[LE/VSE Programming Guide](#)  
[LE/VSE Programming Reference](#)  
[Run-Time Migration Guide, SC33-6687](#)  
[LE/VSE Writing Interlanguage Communication Applications](#)  
[LE/VSE C Run-Time Programming Guide](#)  
[LE/VSE C Run-Time Library Reference](#)

## LE/VSE-Conforming Language Product Publications

---

### **IBM C for VSE/ESA**

[Licensed Program Specifications](#)  
[Installation and Customization Guide](#)  
[Migration Guide](#)  
[User's Guide](#)  
[Language Reference](#)  
[Diagnosis Guide](#)

### **IBM COBOL for VSE/ESA**

[General Information](#)  
[Licensed Program Specifications](#)  
[Migration Guide](#)  
[Installation and Customization Guide](#)  
[Programming Guide](#)  
[Language Reference](#)  
[Diagnosis Guide](#)  
[Reference Summary](#)  
[Millennium Language Extensions Guide](#)

### **IBM PL/I for VSE/ESA**

[Fact Sheet](#)  
[Programming Guide](#)  
[Language Reference](#)  
[Licensed Program Specifications](#)  
[Migration Guide](#)  
[Installation and Customization Guide](#)  
[Diagnosis Guide](#)  
[Compile-Time Messages and Codes](#)  
[Reference Summary](#)

## **Debug Tool for VSE/ESA**

[User's Guide and Reference](#)

[Installation and Customization Guide](#)

[Fact Sheet](#)

## **Softcopy Publications**

---

The following collection kit contains the LE/VSE and LE/VSE-conforming language product publications:

*VSECollection,SK2T-0060*

You can order these publications from Mechanicsburg through your IBM representative.





SC33-6687-00

