

**IBM Library Server Print Preview**

DOCNUM = GC09-2422-00  
DATE TIME = 10/23/96 11:01:02  
BLDVERS = 1.2  
TITLE = C/VSE V1R1 Installation and Customization Guide  
AUTHOR =  
COPYR = © Copyright IBM Corp. 1996  
PATH = /home/webapps/epubs/htdocs/book

---

## COVER Book Cover

---

**IBM C for VSE/ESA**

**Installation and Customization Guide**

Release 1

Document Number GC09-2422-00

Program Number  
5686-A01

---

## NOTICES Notices

**Note!**

Before using this information and the product it supports, be sure to read the general information under ["Notices" in topic FRONT\\_1](#).

---

## EDITION Edition Notice

**First Edition (December 1996)**

This edition applies to Version 1, Release 1, Modification Level 0, of IBM C for VSE/ESA (Program 5686-A01); Version 1, Release 4, Modification Level 0, of IBM Language Environment for VSE/ESA (Program 5686-094); the VSE C Language Run-Time Support feature of VSE/ESA

Version 2 Release 2 (Program 5690-VSE); and to all subsequent releases and modifications until otherwise indicated in new editions. 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 readers' comments is provided at the back of this publication. If the form has been removed, address your comments to:

IBM Canada Ltd. Laboratory  
Information Development  
2G/345/1150/TOR  
1150 Eglinton Avenue East  
North York, Ontario, Canada M3C 1H7

You can also send your comments by facsimile (attention: RCF Coordinator), or you can send your comments electronically to IBM. See "Communicating Your Comments to IBM" for a description of the methods. This page immediately precedes the Readers' Comment Form at the back of this publication.

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 1996. All rights reserved.

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

---

## CONTENTS Table of Contents

### [Summarize](#)

COVER	<u><a href="#">Book Cover</a></u>
NOTICES	<u><a href="#">Notices</a></u>
EDITION	<u><a href="#">Edition Notice</a></u>
CONTENTS	<u><a href="#">Table of Contents</a></u>
TABLES	<u><a href="#">Tables</a></u>
FIGURES	<u><a href="#">Figures</a></u>
FRONT_1	<u><a href="#">Notices</a></u>
FRONT_1.1	<u><a href="#">Programming Interface Information</a></u>
FRONT_1.2	<u><a href="#">Trademarks</a></u>
PREFACE	<u><a href="#">About This Book</a></u>
PREFACE.1	<u><a href="#">The C Language</a></u>
PREFACE.2	<u><a href="#">IBM C for VSE/ESA</a></u>
PREFACE.3	<u><a href="#">IBM Language Environment for VSE/ESA</a></u>
PREFACE.4	<u><a href="#">Using Your Documentation</a></u>
PREFACE.5	<u><a href="#">How to Read the Syntax Diagrams</a></u>
1.0	<u><a href="#">Chapter 1. Planning for C/VSE</a></u>
1.1	<u><a href="#">Planning to Install C/VSE</a></u>

- 1.1.1 [What You Receive with C/VSE](#)
- 1.1.2 [What You Need to Install C/VSE](#)
- 1.1.3 [Planning Where to Install C/VSE](#)
- 1.1.4 [Selecting National Language Support](#)
- 1.1.5 [Checking Service Updates](#)
- 1.1.6 [Publications Useful During Installation](#)
- 1.2 [Planning for Customizing C/VSE](#)
  - 1.2.1 [Deciding Whether to Customize](#)
  - 1.2.2 [Planning to Customize C/VSE Options](#)
  - 1.2.3 [Planning to Install in the Shared Virtual Area](#)
  - 1.2.4 [Planning to Change the National Language for C/VSE](#)
- 2.0 [Chapter 2. Installing C/VSE](#)
- 2.1 [Overview of Installation](#)
  - 2.1.1 [Checklist for Installing C/VSE](#)
  - 2.2 [Step 1: Backup the Original System](#)
  - 2.3 [Step 2: Allocate Space for the Library](#)
    - 2.3.1 [Defining a Library in VSAM Space](#)
    - 2.3.2 [Defining a Library in Non-VSAM Space](#)
  - 2.4 [Step 3: Install C/VSE](#)
    - 2.4.1 [Method 1: Install C/VSE Using the Interactive Interface](#)
    - 2.4.2 [Method 2: Install C/VSE Using a Batch Job](#)
  - 2.5 [Step 4: Verify the Installation of C/VSE](#)
- 3.0 [Chapter 3. Customizing C/VSE](#)
- 3.1 [Customizing Compile-Time Option Defaults](#)
  - 3.1.1 [Modify Default Values of Compile-Time Options](#)
  - 3.1.2 [Link-Edit the C/VSE Compiler Phase EDCDP](#)
- 3.2 [Loading the Compiler into the Shared Virtual Area](#)
- 3.3 [Changing the National Language for C/VSE](#)
- 3.4 [Compile-Time Option Descriptions](#)
  - 3.4.1 [AGGR](#)
  - 3.4.2 [CHECKOUT](#)
  - 3.4.3 [CHKACCUR](#)
  - 3.4.4 [CHKENUM](#)
  - 3.4.5 [CHKEXTRN](#)
  - 3.4.6 [CHKGENERAL](#)
  - 3.4.7 [CHKGOTO](#)
  - 3.4.8 [CHKINIT](#)
  - 3.4.9 [CHKPARAM](#)
  - 3.4.10 [CHKPART](#)
  - 3.4.11 [CHKPPCHK](#)
  - 3.4.12 [CHKPPTRC](#)
  - 3.4.13 [CHKTRUNC](#)
  - 3.4.14 [CSECT](#)
  - 3.4.15 [DECK](#)
  - 3.4.16 [EXECOPS](#)
  - 3.4.17 [EXPMAC](#)
  - 3.4.18 [FLAG](#)
  - 3.4.19 [GONUMB](#)
  - 3.4.20 [HWOPTSTR](#)
  - 3.4.21 [INFILE](#)
  - 3.4.22 [INLNON](#)
  - 3.4.23 [INLNAUT](#)
  - 3.4.24 [INLNREP](#)
  - 3.4.25 [INLNTHR](#)
  - 3.4.26 [INLNLIM](#)
  - 3.4.27 [LANGLVL](#)
  - 3.4.28 [LIST](#)
  - 3.4.29 [LOCALEON](#)
  - 3.4.30 [LOCALE](#)
  - 3.4.31 [LONGNAME](#)
  - 3.4.32 [LSEARCH](#)
  - 3.4.33 [MCOL1F](#)
  - 3.4.34 [MCOL2F](#)
  - 3.4.35 [MCOL1V](#)
  - 3.4.36 [MCOL2V](#)
  - 3.4.37 [MEMORY](#)
  - 3.4.38 [NAME](#)
  - 3.4.39 [NESTINC](#)
  - 3.4.40 [OBJECT](#)

3.4.41	<a href="#">OFFSET</a>
3.4.42	<a href="#">OPT</a>
3.4.43	<a href="#">PPONLY</a>
3.4.44	<a href="#">RENT</a>
3.4.45	<a href="#">SEARCH</a>
3.4.46	<a href="#">SONF</a>
3.4.47	<a href="#">SONV</a>
3.4.48	<a href="#">SCOL1F</a>
3.4.49	<a href="#">SCOL2F</a>
3.4.50	<a href="#">SCOL1V</a>
3.4.51	<a href="#">SCOL2V</a>
3.4.52	<a href="#">SHOWINC</a>
3.4.53	<a href="#">SOURCE</a>
3.4.54	<a href="#">SPILL</a>
3.4.55	<a href="#">SSCOMM</a>
3.4.56	<a href="#">START</a>
3.4.57	<a href="#">TARGET</a>
3.4.58	<a href="#">TERM</a>
3.4.59	<a href="#">TESTON</a>
3.4.60	<a href="#">TESTBLK</a>
3.4.61	<a href="#">TESTLIN</a>
3.4.62	<a href="#">TESTPTH</a>
3.4.63	<a href="#">TESTSYM</a>
3.4.64	<a href="#">UPCONV</a>
3.4.65	<a href="#">XREF</a>
4.0	<a href="#">Chapter 4. Maintaining C/VSE</a>
4.1	<a href="#">Re-Installing C/VSE</a>
4.2	<a href="#">Applying Service Updates</a>
4.2.1	<a href="#">What You Receive</a>
4.2.2	<a href="#">Checklist for Applying Service</a>
4.2.3	<a href="#">Step 1. Check Prerequisite APARs or PTFs</a>
4.2.4	<a href="#">Step 2. Backup Original System</a>
4.2.5	<a href="#">Step 3. Apply Service</a>
4.2.6	<a href="#">Step 4. Run the Installation Verification Program (IVP)</a>
4.3	<a href="#">Removing C/VSE</a>
4.4	<a href="#">How to Report a Problem with C/VSE</a>
BIBLIOGRAPHY	<a href="#">Bibliography</a>
BIBLIOGRAPHY.1	<a href="#">IBM C for VSE/ESA Publications</a>
BIBLIOGRAPHY.2	<a href="#">IBM Language Environment for VSE/ESA Publications</a>
BIBLIOGRAPHY.3	<a href="#">Related Publications</a>
BIBLIOGRAPHY.4	<a href="#">Softcopy Publications</a>
INDEX	<a href="#">Index</a>
BACK_1	<a href="#">Communicating Your Comments to IBM</a>
COMMENTS	<a href="#">Readers' Comments -- We'd Like to Hear from You</a>

---

## TABLES Tables

1.	<a href="#">How to Use C/VSE Publications</a>	PREFACE.4
2.	<a href="#">How to Use LE/VSE Publications</a>	PREFACE.4
3.	<a href="#">Component ID and CLC</a>	1.1.1
4.	<a href="#">Basic material: VSE/ESA Optional Program Tape</a>	1.1.1.1
5.	<a href="#">Basic material: Separately-Orderable Distribution Tape</a>	1.1.1.1
6.	<a href="#">Distribution media: file contents</a>	1.1.1.1
7.	<a href="#">Basic Material: Unlicensed Publications</a>	1.1.1.2
8.	<a href="#">Required Licensed Programs for C/VSE</a>	1.1.2.1
9.	<a href="#">Optional Licensed Programs for C/VSE</a>	1.1.2.1
10.	<a href="#">Approximate Library Storage Requirements for C/VSE</a>	1.1.2.2
11.	<a href="#">Compile-Time Option Defaults</a>	1.2.2
12.	<a href="#">SVA Space Requirements for C/VSE Phases</a>	1.2.3
13.	<a href="#">Summary of steps for installing C/VSE</a>	2.1.1
14.	<a href="#">Summary of steps for installing service on C/VSE</a>	4.2.2

---

# FIGURES Figures

- [1. Define the C/VSE Library Cluster in VSAM Space. 2.3.1](#)
  - [2. Define the LE/VSE Library. 2.3.1](#)
  - [3. List the Contents of Volume SYSWK1. 2.3.2](#)
  - [4. Define the LE/VSE Library. 2.3.2](#)
  - [5. Install C/VSE 2.4.2](#)
  - [6. Installation Verification 2.5](#)
  - [7. Example Output from EDCXIVP2 2.5](#)
  - [8. Job to assemble and catalog the default compile-time options 3.1.1](#)
  - [9. Job to link-edit the default options phase 3.1.2](#)
  - [10. Example of the PARM option with NATLANG 3.3](#)
  - [11. Retrace APARS and PTFs 4.2.3](#)
  - [12. Apply Service 4.2.5.2](#)
  - [13. Delete C/VSE from a Sublibrary 4.3](#)
  - [14. Remove C/VSE from the System History File 4.3](#)
- 

## FRONT\_1 Notices

Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

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, USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independent created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Canada Ltd., Department 071, 1150 Eglinton Avenue East, North York, Ontario M3C 1H7, Canada. Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

Subtopics:

- [FRONT\\_1.1 Programming Interface Information](#)
  - [FRONT\\_1.2 Trademarks](#)
- 

## FRONT\_1.1 Programming Interface Information

This book is intended to help you to plan for, install, customize, and maintain IBM C for VSE/ESA. This book documents the procedures required to install, customize, and maintain IBM C for VSE/ESA.

---

## FRONT\_1.2 Trademarks

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

AD/Cycle	IBM	SAA
AIX/6000	Language Environment	SQL/DS
BookManager	OS/2	VSE/ESA
C/370	OS/390	
CICS/VSE	OS/400	
DFSORT	QMF	

The following terms are trademarks of other companies:

ANSI	American National Standards Institute
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers
ISO	International Organization for Standardization
POSIX	Institute of Electrical and Electronic Engineers

---

## PREFACE About This Book

This book is intended for users of IBM C for VSE/ESA (C/VSE) as implemented for the IBM Language Environment for VSE/ESA (LE/VSE) environment. It contains information on planning for, installing, and customizing C/VSE.

**Note:** References to LE/VSE also apply to the VSE C Language Run-Time Support feature of VSE/ESA Version 2 Release 2.

Subtopics:

- [PREFACE.1 The C Language](#)
- [PREFACE.2 IBM C for VSE/ESA](#)
- [PREFACE.3 IBM Language Environment for VSE/ESA](#)
- [PREFACE.4 Using Your Documentation](#)
- [PREFACE.5 How to Read the Syntax Diagrams](#)

---

### PREFACE.1 The C Language

The C language is a general purpose, function-oriented programming language that allows a programmer to create applications quickly and easily. C provides high-level control statements and data types as do other structured programming languages, and it also provides many of the benefits of a low-level language. Using the C/VSE language, you can write portable code conforming to the ANSI standard.

IBM offers the C language on other platforms, such as the OS/2, AIX/6000, OS/400, OS/390, and VM operating systems.

The elements of the C/VSE implementation include:

- All elements of the joint ISO and IEC standard: ISO/IEC 9899:1990 (E)
- ANSI/ISO 9899:1990[1992] (formerly ANSI X3.159-1989 C)
- Locale based internationalization support as defined in: ISO/IEC DIS 9945-2:1992/IEEE POSIX 1003.2-1992 Draft 12 (There are some limitations to fully-compliant behavior as noted in the *LE/VSE C Run-Time Programming Guide*.)
- Extended multibyte and wide character utilities as defined by a subset of the Programming Language C Amendment 1, which will be ISO/IEC 9899:1990/Amendment 1:1994(E)

---

## PREFACE.2 IBM C for VSE/ESA

This is the first release of C/VSE. It brings to the VSE environment many of the functions of IBM SAA AD/Cycle C/370 Version 1 Release 2, while retaining source compatibility with IBM C/370 for VSE.

C/VSE enables you to run your C applications with LE/VSE. In addition to C's already impressive features, you gain access to LE/VSE's rich set of library routines and enhanced interlanguage communication (ILC) with other LE/VSE-conforming languages.

Some of the features of C/VSE include:

*Interlanguage communication (ILC) support:*

- Object code produced by C/VSE can be linked with object code produced by other LE/VSE-conforming compilers, IBM COBOL for VSE/ESA (COBOL/VSE) and IBM PL/I for VSE/ESA (PL/I VSE).
- C/VSE programs can fetch COBOL/VSE or PL/I VSE phases.
- COBOL/VSE or PL/I VSE programs can fetch C/VSE phases.

**Note:** C/VSE does not support ILC with:

- FORTRAN
- RPG
- DOS/VS COBOL
- C/370

Limited ILC support is provided for VS COBOL II at Release 3.2 or later.

*Debugging:*

C/VSE provides support for Debug Tool for VSE/ESA to assist in the debugging of application programs.

*Usability:*

- International coded character set (codepage) support. The compiler can process special language specific characters in source files without the use of alternative characters.
- Fixed-point decimal (packed decimal) data as a native C data type. The C/VSE fixed-point decimal type allows expressions with up to 31 digits of precision.

*Product packaging:*

All C/VSE library routines are packaged with LE/VSE, the dynamic routines are loaded at run time and not link-edited with the application program. Changes to the dynamic library do not require C programs to be re-linked.

For migration considerations, see the [C/VSE Migration Guide](#).

---

## **PREFACE.3 IBM Language Environment for VSE/ESA**

C/VSE exploits the C run-time environment and library of run-time callable services provided by IBM Language Environment for VSE/ESA (LE/VSE).

LE/VSE establishes a common run-time environment and common run-time callable services for language products, user programs, and other products.

The common execution environment is made up of data items and services performed by library routines available to a particular application running in the environment. The services that LE/VSE provides to your application may include:

- Services that satisfy basic requirements common to most applications. These include support for the initialization and termination of applications, allocation of storage, support for interlanguage communication (ILC), and condition handling.
- Extended services often needed by applications. These functions are contained within a library of callable routines, and include interfaces to operating system functions and a variety of other commonly used functions.



- Run-time options that help the execution, performance tuning, performance, and diagnosis of your application.
- Access to language-specific library routines, such as the C functions.

## PREFACE.4 Using Your Documentation

The publications in the C/VSE and LE/VSE libraries are designed to help you develop C/VSE applications that run with LE/VSE. Each publication helps you perform a different task. For a complete list of publications you might need, see ["Bibliography" in topic BIBLIOGRAPHY](#). [Table 1](#) lists the publications in the C/VSE library.

To...	Use...	
Plan for, install, customize, and maintain C/VSE	<i>Installation and Customization Guide</i>	GC09-2422
Migrate VSE applications from C/370 to C/VSE	<a href="#">Migration Guide</a>	SC09-2423
Get details on C/VSE syntax and specifications of language elements	<a href="#">Language Reference</a>	SC09-2425
Find syntax for compile-time options; compile your C/VSE applications; get details on compile-time messages	<a href="#">User's Guide</a>	SC09-2424
Diagnose compiler problems and report them to IBM	<a href="#">Diagnosis Guide</a>	GC09-2426
Understand warranty information	<i>Licensed Program Specifications</i>	GC09-2421

[Table 2](#) lists the publications in the LE/VSE library. These include publications designed to help you develop and debug your C/VSE applications, diagnose run-time problems that occur in your C/VSE applications, and use C/VSE-related utilities.

To...	Use...	
Evaluate LE/VSE	<i>Fact Sheet</i> <i>Concepts Guide</i>	GC33-6679 GC33-6680
Plan for, install, customize, and maintain LE/VSE	<i>Installation and Customization Guide</i>	SC33-6682
Understand the LE/VSE program models and concepts	<i>Concepts Guide</i> <i>Programming Guide</i>	GC33-6680 SC33-6684
Find syntax for LE/VSE run-time options and callable services	<i>Programming Reference</i>	SC33-6685
Develop your C/VSE applications	<i>Programming Guide</i> <i>C Run-Time Programming Guide</i> <i>C Run-Time Library Reference</i>	SC33-6684 SC33-6688 SC33-6689

Develop interlanguage communication (ILC) applications	<i>Writing Interlanguage Communication Applications</i>	SC33-6686
Debug your C/VSE applications and get details on run-time messages	<i>Debugging Guide and Run-Time Messages</i>	SC33-6681
Migrate applications to LE/VSE	<i>Run-Time Migration Guide</i>	SC33-6687
Diagnose run-time problems that occur in your C/VSE applications	<i>Debugging Guide and Run-Time Messages</i>	SC33-6681
Use C/VSE-related utilities	<i>C Run-Time Programming Guide</i>	SC33-6688
Understand warranty information	<i>Licensed Program Specifications</i>	GC33-6683

## PREFACE.5 How to Read the Syntax Diagrams

In this book, syntax for commands, directives, and statements is described using the following structure:

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line.

A double right-arrowhead indicates the beginning of a command, directive, or statement; the single right-arrowhead indicates that it is continued on the next line. (In the following diagrams, statement is used to represent a command, directive, or statement.)

```
>>__statement__>
```

The following indicates a continuation; the opposing arrowheads indicate the end of a command, directive, or statement.

```
>__statement__<
```

Diagrams of syntactical units other than complete commands, directives, or statements look like this:

```
>__statement__>
```

- Required items are on the horizontal line (the main path).

```
>>__statement__required_item__<
```

- IBM-supplied default items are above the main path.

```
>>__statement__|_____|__<
```

- Optional items are below the main path.

```
>>__statement__ _____ __<
```

|\_optional\_item\_|

- If you can choose from two or more items, they are vertical in a stack.

If you *must* choose one of the items, one item of the stack is on the main path.

```
>>__statement__  _required_choice1_  __<
                    |_required_choice2_|
```

If choosing one of the items is optional, the entire stack is below the main path.

```
>>__statement__  _____  __<
                    |_optional_choice1_|
                    |_optional_choice2_|
```

- An arrow returning to the left above a line indicates an item that you can repeat.

```
>>__statement__  <_____
                    _repeatable_item_|  __<
```

or

```
>>__statement__  _____  __<
                    | <_____ |
                    |__repeatable_item_|_|
```

A repeat arrow above a stack indicates that you can make more than one choice from the stacked items, or repeat a single choice.

- Keywords are in non-italic letters and should be entered exactly as shown (for example, `pragma`). They must be spelled exactly as shown. Variables are in italics and lowercase letters (for example, *identifier*). They represent user-supplied names or values.
- If punctuation marks, parentheses, arithmetic operators, or other non-alphanumeric characters are shown, you must enter them as part of the syntax.

**Note:** The white space is not always required between tokens but you should include at least one blank space between tokens unless otherwise specified.

The following syntax diagram example shows the syntax for the `#pragma` comment directive.

```
>>_#_____pragma_____comment_____(_____)>
      (1,2)          (3)          (4)  (5)

>_  _____(_____)_____><
   (6)                                     (9,10)
|_date_____|
|_timestamp_____|
```

```
|_ _copyright_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ |  
|_user_____ | | (7) (8) |  
|_,_____ "characters" _____|
```

**Notes:**

- (1) This is the start of the syntax diagram.
- (2) The symbol # must appear first.
- (3) The keyword pragma must follow the # symbol.
- (4) The keyword comment must follow the keyword pragma.
- (5) An opening parenthesis must follow the keyword comment.
- (6) The comment type must be entered only as one of the following: compiler, date, timestamp, copyright, or user.
- (7) If the comment type is copyright or user, and an optional character string is following, a comma must be present after the comment type.
- (8) A character string must follow the comma. The character string must be enclosed in double quotation marks.
- (9) A closing parenthesis is required.
- (10) This is the end of the syntax diagram.

The following examples of the #pragma comment directive are syntactically correct according to the diagram shown above:

```
#pragma comment(date)  
#pragma comment(user)  
#pragma comment(copyright,"This text will appear in the module")
```

---

## 1.0 Chapter 1. Planning for C/VSE

This chapter helps you plan for installing and customizing C/VSE.

Subtopics:

- [1.1 Planning to Install C/VSE](#)
- [1.2 Planning for Customizing C/VSE](#)

---

### 1.1 Planning to Install C/VSE

This section contains planning information to help you properly install C/VSE:

- What you receive with C/VSE
- What you need to install C/VSE
- Planning where to install C/VSE
- Selecting national language support
- Checking service updates
- Publications useful during installation

Subtopics:

- [1.1.1 What You Receive with C/VSE](#)
- [1.1.2 What You Need to Install C/VSE](#)
- [1.1.3 Planning Where to Install C/VSE](#)
- [1.1.4 Selecting National Language Support](#)
- [1.1.5 Checking Service Updates](#)
- [1.1.6 Publications Useful During Installation](#)

## 1.1.1 What You Receive with C/VSE

You can receive C/VSE in one of two forms:

- As an optional program on the VSE/ESA optional program tape
- As a separately-orderable licensed program on a separate distribution tape

When you order C/VSE, either as an optional program, or as a separately-orderable licensed program, you receive:

- Basic machine-readable material
- Basic unlicensed publications

[Table 3](#) lists the component identifiers (COMP IDs) and component level codes (CLCs) for each component of C/VSE. You can install either both the C/VSE components you received, or just the C/VSE base component.

Component ID	CLC	Description
5686-A01-00	1FY	C/VSE Base with uppercase and mixed-case U.S. English
5686-A01-01	1G4	C/VSE Japanese NLF

The following sections describe these materials and publications.

Subtopics:

- [1.1.1.1 Distribution Media](#)
- [1.1.1.2 Program Documentation](#)

### 1.1.1.1 Distribution Media

You receive C/VSE on one of the following:

- 9-track magnetic tape written at 6250 BPI
- IBM 3480 or IBM 3490 cartridge
- 4-mm DAT cartridge

The cartridge or tape contains all the programs and data you need to install C/VSE.

*Basic Machine-Readable Material:* [Table 4](#) describes the VSE/ESA optional program tape that contains C/VSE. [Table 5](#) describes the separately-orderable licensed program distribution tape.

Depending on the distribution medium feature you ordered, you will receive basic machine-readable material on one of the media described in these tables. To simplify installation, the C/VSE distribution media includes uppercase U.S. English (UEN), mixed-case U.S. English (ENU), and the Japanese (JPN) national language feature, regardless of whether you ordered the Japanese national language feature (NLF).

Medium	Feature number	Physical volume	External label	Volser
6250 tape	3401	1	VSE/ESA 2.2.0 OPTIONAL PROGRAMS n OF x	unlabeled
IBM 3480 and IBM 3490	3405	1	VSE/ESA 2.2.0 OPTIONAL PROGRAMS n OF x	unlabeled
4-mm DAT cartridge	3431	1	VSE/ESA 2.2.0 OPTIONAL PROGRAMS n OF x	unlabeled

Medium	Feature number	Physical volume	External label	Volser
6250 tape	5975 5988 5260 5263	1	C/VSE COMPILER	unlabeled
IBM 3480 and IBM 3490	5976 5989 5261 5264	1	C/VSE COMPILER	unlabeled
4-mm DAT cartridge	6300 6301	1	C/VSE COMPILER	unlabeled

	5262			
	5265			

The file content of the distribution tape you receive depends upon the form in which you receive C/VSE. If you receive C/VSE on the VSE/ESA optional program tape, there might be other licensed programs on the tape. However, if you ordered C/VSE separately, the distribution tape will contain only C/VSE. [Table 6](#) describes the file contents of the separately-orderable licensed program distribution tape.

Table 6. Distribution media: file contents	
File	Contents
1	Header file containing C/VSE copyright statement
2	Backup file ID C/VSE.BASE.1.1.0 followed by an MSHP System History File
3	C/VSE library file containing the C/VSE product with uppercase and mixed-case U.S. English
4	Header file containing the C/VSE copyright statement
5	Backup file ID C/VSE.JPN..1.1.0 followed by an MSHP System History File
6	C/VSE library file containing the C/VSE Japanese NLF
7	Null (Tape Mark)
8	End of backup (EOB) record
9	Null (Tape Mark)

*Optional Material:* There are no optional machine-readable materials for C/VSE.

### 1.1.1.2 Program Documentation

This section identifies the basic and optional C/VSE documentation you receive.

*Basic Unlicensed Publications:* [Table 7](#) identifies the basic publications for C/VSE. Unless otherwise noted, you receive one of each of these publications when you receive the basic materials for C/VSE. For additional copies, contact your IBM representative.

Table 7. Basic Material: Unlicensed Publications	
Publication title	Form number
C/VSE Licensed Program Specifications	GC09-2421( <a href="#">1</a> )

C/VSE Installation and Customization Guide	GC09-2422( <a href="#">1</a> )
C/VSE Migration Guide	SC09-2423( <a href="#">1</a> )
C/VSE User's Guide	SC09-2424( <a href="#">1</a> )
C/VSE Language Reference	SC09-2425( <a href="#">1</a> )
C/VSE Diagnosis Guide	GC09-2426( <a href="#">1</a> )
<b>Note:</b>	
1. Add "N:" before each form number (for example N:GC09-2422) to specify publications in Japanese.	

*Optional Unlicensed Program Publications:* There are no optional unlicensed program publications for C/VSE.

For a list of publications for related products, see ["Bibliography" in topic BIBLIOGRAPHY](#).

---

## 1.1.2 What You Need to Install C/VSE

The following sections describe system requirements for installing C/VSE.

Subtopics:

- [1.1.2.1 Licensed Programs](#)
- [1.1.2.2 DASD Storage](#)

---

### 1.1.2.1 Licensed Programs

C/VSE runs under VSE/ESA with the required licensed programs listed in [Table 8](#) and optional licensed programs listed in [Table 9](#). You should install all licensed programs with the minimum release listed or with any subsequent release.

The licensed programs listed in [Table 8](#) are required to install and customize C/VSE, or to run C/VSE.

Required Licensed Program	Minimum Release	Program Number
One of:		
VSE/ESA	Version 1 Release	5750-ACD
VSE/ESA	4	5690-VSE
	Version 2 Release	
	1	



One of: IBM Language Environment for VSE/ESA including the C-specific base component  VSE C Language Run-Time Support feature of VSE/ESA Version 2 Release 2	Version 1 Release 4	5686-094  5690-VSE
High Level Assembler/MVS & VM & VSE	Release 1	5696-234

The licensed programs listed in [Table 9](#) can optionally be used with C/VSE.

Optional Licensed Program	Minimum Release	Program Number
CICS/VSE	Version 2 Release 3	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
BookManager/Read	Release 2 (to view softcopy documentation)	73F6-023 (Read/2)
IBM COBOL for VSE/ESA	Release 1	5686-068
IBM DL/I DOS/VS	Release 10	5746-XX1
IBM PL/I for VSE/ESA	Release 1	5686-069
IBM DOS/VS Sort/Merge	Version 2 Release 5	5746-SM2
QMF/VSE	Version 3 Release 2	5648-061
SQL/DS	Version 3 Release 4	5688-103

Debug Tool for VSE/ESA, which is an optional feature of the LE/VSE-conforming high-level language compilers, can also be used with C/VSE.

### 1.1.2.2 DASD Storage

When installing C/VSE you must provide DASD storage for the VSE Librarian library.

[Table 10](#) lists the estimated minimum DASD storage required for VSE Librarian library storage. The storage requirements given in [Table 10](#) are in addition to the storage requirements of any other licensed programs installed in the library.

Due to the size of the phases in the C/VSE sublibrary, you should consider allowing 75% to 100% extra library storage for any future enhancements and service updates.

Table 10. Approximate Library Storage Requirements for C/VSE					
Component	LIBR BLKS(1)	3380 TRK	3390 TRK	9345 TRK	FBA BLKS
C/VSE base	6500	215	185	215	13000
C/VSE Japanese NLF	200	8	7	8	400
Total	6700	223	192	223	13400

**Note:**

- One (1) library block equals 1024 bytes.

### 1.1.3 Planning Where to Install C/VSE

The installation procedures described in [Chapter 2, "Installing C/VSE" in topic 2.0](#) assume that all components of C/VSE are to be installed in the DBASE sublibrary of the PRD2 library. You can choose to install C/VSE into any other library and sublibrary, with the exception that C/VSE must **not** be installed in the same sublibrary as the C/370 compiler because of phase and module name conflicts.

When you have decided what library and sublibrary to use, make sure the library has sufficient free space to hold the C/VSE licensed program. [Table 10 in topic 1.1.2.2](#) lists the minimum library storage requirements.

You can check the number of free library blocks in a library using the LIBR program with the LISTDIR command. For more information about the LIBR program, see *VSE/ESA System Control Statements*.

### 1.1.4 Selecting National Language Support

Uppercase and mixed-case U.S. English are both included in the base component for C/VSE. You do not have to install the mixed-case U.S. English as a separate component. The Japanese national language feature is included in a separate component, that you need to install if you require Japanese.

When installing C/VSE, you cannot choose the default national language for

the compiler messages and heading lines. C/VSE runs in an LE/VSE environment and therefore the national language default for C/VSE is the same as the national language default for LE/VSE. If you need to change the national language used for a particular execution of the compiler, see ["Changing the National Language for C/VSE" in topic 3.3.](#)

**Note:** If your default national language for LE/VSE is Japanese you **must** install the Japanese national language feature of C/VSE.

---

## 1.1.5 Checking Service Updates

Before installing C/VSE, check with your IBM Support Center or use either Information/Access or SoftwareXcel Extended to see whether there is additional service information you need. To get this information, specify the following:

<b>UPGRADE</b>	CVSE110
<b>SUBSET</b>	A011FY
	A011G4

---

## 1.1.6 Publications Useful During Installation

For a list of VSE/ESA publications you might find useful during installation, see ["Bibliography" in topic BIBLIOGRAPHY.](#)

---

# 1.2 Planning for Customizing C/VSE

This section contains planning information to help you customize C/VSE:

- Deciding whether to customize
- Planning to customize C/VSE options
- Planning to install in the Shared Virtual Area (SVA)
- Planning to change the national language for C/VSE

Subtopics:

- [1.2.1 Deciding Whether to Customize](#)
  - [1.2.2 Planning to Customize C/VSE Options](#)
  - [1.2.3 Planning to Install in the Shared Virtual Area](#)
  - [1.2.4 Planning to Change the National Language for C/VSE](#)
- 

## 1.2.1 Deciding Whether to Customize

You should consider whether the IBM-supplied values for the C/VSE compile-time options suit the needs of your site. Compile-time options control features such as:

- Suppressing part of the listing
- Generating symbol cross-reference information

You should also consider whether to load the compiler phases in the SVA, and whether you need to change the default for the national language.

Make sure C/VSE serves the needs of the application programmers at your site. Confer with them while you evaluate the customization options for C/VSE, particularly those options that are also available to the application programmers. This ensures the modifications you make best support the application programs being developed at your site.

## 1.2.2 Planning to Customize C/VSE Options

[Table 11](#) lists the IBM-supplied default values for the options in the C/VSE customization assembler macro. The options in this assembler macro determine the defaults for C/VSE compile-time options. If you plan to customize a default, write the new value in the "Customized Installation Default" column.

These options are described in more detail in [Chapter 3, "Customizing C/VSE" in topic 3.0](#).

Option	See Note	Customized Installation Default	IBM-Supplied Default	Description
>>__AGGR=_ <u>  </u> _ <u>  </u> _ <u>  </u> ><		_____	0	Lists structures, unions, and their sizes
>>__CHECKOUT=_ <u>  </u> _ <u>  </u> _ <u>  </u> ><		_____	0	Gives informational messages about possible programming errors (see options marked " <a href="#">(1)</a> " in the Note column)
>>__CHKACCUR=_ <u>  </u> _ <u>  </u> _ <u>  </u> ><	<a href="#">(1)</a>	_____	1	Enables messages for assignments of long values to variables that are not long
>>__CHKENUM=_ <u>  </u> _ <u>  </u> _ <u>  </u> ><	<a href="#">(1)</a>	_____	0	Enables messages for usage of enumerations
>>__CHKEXTRN=_ <u>  </u> _ <u>  </u> _ <u>  </u> ><	<a href="#">(1)</a>	_____	0	Enables messages for never used variables that have external declarations
>>__CHKGENERAL=_ <u>  </u> _ <u>  </u> _ <u>  </u> ><	<a href="#">(1)</a>	_____	1	Enables general checkout messages
>>__CHKGOTO=_ <u>  </u> _ <u>  </u> _ <u>  </u> ><	<a href="#">(1)</a>	_____	1	Enables messages for the appearance and usage of goto statements
				Enables messages for variables

>>__CHKINIT=_ <u>1</u>   <u>0</u>   ___><	( <u>1</u> )	_____	1	that are not explicitly initialized
>>__CHKPARAM=_ <u>1</u>   <u>0</u>   ___><	( <u>1</u> )	_____	1	Enables messages for function parameters that are not used
>>__CHKPORT=_ <u>0</u>   <u>1</u>   ___><	( <u>1</u> )	_____	0	Enables messages for non-portable use of the C language
>>__CHKPPCHK=_ <u>1</u>   <u>0</u>   ___><	( <u>1</u> )	_____	1	Enables messages for all preprocessor directives
>>__CHKPPTRC=_ <u>0</u>   <u>1</u>   ___><	( <u>1</u> )	_____	0	Enables messages for the tracing of include files by the preprocessor
>>__CHKTRUNC=_ <u>1</u>   <u>0</u>   ___><	( <u>1</u> )	_____	1	Enables messages for variable names that are truncated by the compiler
>>__CSECT=_ <u>0</u>   <u>1</u>   ___><		_____	0	Ensures a program contains named CSECTS
>>__EXECOPS=_ <u>1</u>   <u>0</u>   ___><		_____	1	Accept run-time options on the command line
>>__EXPMAC=_ <u>0</u>   <u>1</u>   ___><		_____	0	Shows all expanded macros in the source listing
>>__FLAG=_ <u>0</u>   <u>10</u>   ___><   <u>30</u>     <u>40</u>		_____	0	Lists diagnostic messages of certain types
>>__GONUMB=_ <u>0</u>   <u>1</u>   ___><		_____	0	Generates line number tables
>>__HWOPTSTR=_ <u>0</u>   <u>1</u>   ___><		_____	0	Generates code to take advantage of different hardware
>>__INLNON=_ <u>0</u>   <u>1</u>   ___><		_____	0	Places code for function code at the point of call (see options marked "( <u>2</u> )" in the Note column)
>>__INLNAUT=_ <u>1</u>   <u>0</u>   ___><	( <u>2</u> )	_____	1	Generates automatic auto-inlining of functions.
>>__INLNREP=_ <u>0</u>   <u>1</u>   ___><	( <u>2</u> )	_____	0	Generates an inline report as part of the listing
>>__INLNTHR=_ <u>250</u>   <u>n</u>   ___><	( <u>2</u> )	_____	250	Specifies the maximum relative size of a function to inline
>>__INLNLIM=_ <u>1000</u>   <u>n</u>   ___><	( <u>2</u> )	_____	1000	Specifies the maximum relative size to which a function can grow before auto-inlining stops

>>__LANGLVL=_ <u>  4  </u>  _1_ ____><  _2_ ____><  _3_ ____><	_____	4	Specifies the language standard for the compiler
>>__LOCALEON=_ <u>  0  </u>  _1_ ____><	_____	0	Sets the locale to be used by the compiler
>>__LOCALE=_ <u>          </u>  _name_ ____><	_____	-	Specifies the locale name when LOCALEON=1
>>__LONGNAME=_ <u>  0  </u>  _1_ ____><	_____	0	Specifies the long name support is used
>>__MCOL1F=_ <u>  1          </u>  _left-margin_ ____><	_____	1	Specifies the margin for the first column of the source input for fixed-length files
>>__MCOL2F=_ <u>  72          </u>  _right-margin_ ____><	_____	72	Specifies the margin for the last column of the source input for fixed-length files
>>__MCOL1V=_ <u>  1          </u>  _left-margin_ ____><	_____	1	Specifies the margin for the first column of the source input for variable-length files
>>__MCOL2V=_ <u>  32767          </u>  _right-margin_ ____><	_____	32767	Specifies the margin for the last column of the source input for variable-length files
>>__MEMORY=_ <u>  0  </u>  _1_ ____><	_____	0	Determines if MEMORY files are used for work files
>>__NESTINC=_ <u>  16  </u>  _n_ ____><	_____	16	Determines the number of nested include files allowed
>>__OFFSET=_ <u>  0  </u>  _1_ ____><	_____	0	Determines if the offset address of each function is displayed in the listing
>>__OPT=_ <u>  0  </u>  _1_ ____><	_____	0	Determines the level of compiler optimization
>>__PPONLY=_ <u>  0  </u>  _1_ ____><	_____	0	Determines whether only the preprocessor runs against the source file
>>__RENT=_ <u>  0  </u>  _1_ ____><	_____	0	Determines whether the compiler generates reentrant code
>>__SONF=_ <u>  1          </u>  _0_ ____><	_____	1	Determines whether fixed-length records contain sequence numbers
>>__SONV=_ <u>  0          </u>  _1_ ____><	_____	0	Determines whether variable-length records contain sequence numbers
>>__SCOL1F=_ <u>  73          </u>  _left-margin_ ____><	_____	73	Specifies the left-hand margin of the sequence numbers for fixed-length records

>>__SCOL2F=_ <u>  80  </u>  __><			80	Specifies the right-hand margin of the sequence numbers for fixed-length records
>>__SCOL1V=_ <u>  0  </u>  __><			0	Specifies the left-hand margin of the sequence numbers for variable-length records
>>__SCOL2V=_ <u>  0  </u>  __><			0	Specifies the right-hand margin of the sequence numbers for variable-length records
>>__SHOWINC=_ <u>  0  </u>  __><			0	Shows all include files processed in the compiler listing
>>__SPILL=_ <u> 128 </u>  __><			128	Sets the size of the spill area
>>__SSCOMM=_ <u>  0  </u>  __><			0	Recognize two slashes (//) as a comment line
>>__TESTON=_ <u>  0  </u>  __><			0	Specifies whether the compiler generates information for Debug Tool for VSE/ESA (see options marked "(3)" in the Note column)
>>__TESTBLK=_ <u>  1  </u>  __><	(3)		1	Determines whether the compiler generates symbol information for nested blocks
>>__TESTLIN=_ <u>  1  </u>  __><	(3)		1	Determines whether the compiler generates line number hooks
>>__TESTPTH=_ <u>  0  </u>  __><	(3)		0	Determines whether the compiler generates path break points
>>__TESTSYM=_ <u>  1  </u>  __><	(3)		1	Determines whether the compiler generates symbol table information
>>__UPCONV=_ <u>  0  </u>  __><			0	Specifies whether the compiler follows <i>unsignedness preserving</i> rules when doing C type conversions.

**Notes:**

1. These options are suboptions of CHECKOUT and only effect the output listing when CHECKOUT=1, or when CHECKOUT is coded either in the source file on the #pragma directive or in the command line.
2. These options are suboptions of INLNON and only take effect when INLNON=1, or when INLINE is coded either in the source file on the #pragma directive or in the command line.
3. These options are suboptions of TESTON and only take effect when TESTON=1, or when TEST is coded either in the source file on the #pragma directive or in the command line.

See the [C/VSE User's Guide](#) for a list of options available to the programmer.

### 1.2.3 Planning to Install in the Shared Virtual Area

By placing your C/VSE routines in the shared virtual area (SVA) you will:

- Reduce the overall system storage requirements by making the routines shareable
- Reduce the initialization/termination time for each compiler invocation, because phase load time decreases

Choose which routines to put in the SVA. [Table 12](#) provides an estimate of the amount of space used by each phase that can be put in the SVA. All phases can be put in the 31-bit SVA. The supplied SVA loadlist \$SVAEDCC contains all of the phases listed in [Table 12](#).

Phase	Space Needed for 31-bit SVA (bytes decimal)
EDCCOMP	8,744
EDCDO	583,912
EDCDP	1,162,808
EDCDR	93,280
EDCDT	641,472
Total	2,490,216

### 1.2.4 Planning to Change the National Language for C/VSE

C/VSE executes in an LE/VSE environment and therefore the default national language for C/VSE is the same as the default national language for LE/VSE. You can change the national language used for a particular execution of C/VSE using the LE/VSE run-time option NATLANG. For details see ["Changing the National Language for C/VSE" in topic 3.3](#).

## 2.0 Chapter 2. Installing C/VSE

This chapter describes the installation method and the step-by-step procedures you use to install and activate the functions of C/VSE.



Subtopics:

- [2.1 Overview of Installation](#)
  - [2.2 Step 1: Backup the Original System](#)
  - [2.3 Step 2: Allocate Space for the Library](#)
  - [2.4 Step 3: Install C/VSE](#)
  - [2.5 Step 4: Verify the Installation of C/VSE](#)
- 

## 2.1 Overview of Installation

You install C/VSE using the Maintain System History Program (MSHP), or by using the VSE/ESA Interactive Interface.

Subtopics:

- [2.1.1 Checklist for Installing C/VSE](#)
- 

### 2.1.1 Checklist for Installing C/VSE

[Table 13](#) lists the steps and associated jobs for installing C/VSE. The remaining sections in this chapter describe each step. You can use [Table 13](#) as a checklist.

Step	Description	Installation job	Topic
__ 1	Backup the original system	--	<a href="#">2.2</a>
__ 2	Allocate space for the library (Omit if using the default sublibrary)		<a href="#">2.3</a>
	In VSAM Space	EDCXIVDF EDCXILDF	<a href="#">2.3.1</a>
	In Non-VSAM Space	EDCXINDF	<a href="#">2.3.2</a>
__ 3	Install C/VSE		<a href="#">2.4</a>
	Method 1. Install Debug Tool for VSE/ESA using the Interactive Interface	--	<a href="#">2.4.1</a>
	Method 2. Install Debug Tool for VSE/ESA using a batch job	EDCXINST	<a href="#">2.4.2</a>
__ 4	Verify the installation of	EDCXIVP	<a href="#">2.5</a>

	C/VSE		
--	-------	--	--

---

## 2.2 Step 1: Backup the Original System

Make a backup copy of your current C/VSE library or the library you intend to install C/VSE into, and the system history file.

For information about backing up libraries and the system history file, see *VSE/ESA System Control Statements*.

---

## 2.3 Step 2: Allocate Space for the Library

By default, C/VSE is installed into the PRD2.DBASE sublibrary. If you want to install C/VSE into the default sublibrary, go to ["Step 3: Install C/VSE" in topic 2.4](#). If you want to install C/VSE into a sublibrary other than PRD2.DBASE, then continue with this step.

Decide whether you want to install C/VSE into a library defined in VSAM space or in non-VSAM space.

If you want to install C/VSE into a library defined in VSAM space, you need to:

1. Define the VSAM cluster for the library using the VSAM utility IDCAMS.
2. Define the library and sublibrary using the VSE Librarian utility LIBR.
3. If the library is not already defined in your standard labels or in your label procedures, add the necessary label information to the appropriate procedure.

See ["Defining a Library in VSAM Space" in topic 2.3.1](#) for a sample job to define a library in VSAM space.

If you want to install C/VSE into a library defined in non-VSAM space, you need to:

1. Decide where to allocate space for the C/VSE library.
2. Define the library and sublibrary using the VSE Librarian utility LIBR.
3. If the library is not already defined in your standard labels or in your label procedures, add the necessary label information to the appropriate procedure.

See ["Defining a Library in Non-VSAM Space" in topic 2.3.2](#) for sample job control to define a library in non-VSAM space.

Subtopics:

- [2.3.1 Defining a Library in VSAM Space](#)
- [2.3.2 Defining a Library in Non-VSAM Space](#)

## 2.3.1 Defining a Library in VSAM Space

The sample job in [Figure 1](#) defines a library in VSAM space. You will need to tailor some statements to suit your installation. The tailoring requirements for each statement are discussed in the notes following [Figure 1](#).

```
// JOB  EDCXIVDF
*
*   Define the C/VSE VSAM cluster
*
// OPTION LOG
*           _____ 1
*
// DLBL  IJSYSUC, 'VSESP.USER.CATALOG', ,VSAM
// EXEC  IDCAMS,SIZE=AUTO
           _____ 2

DELETE (CVSE.LIBRARY) CLUSTER PURGE
DEFINE CLUSTER - "
           ___|_____ 3

           (NAME(CVSE.LIBRARY) -
           VOLUME(volser) - _____ 4
           NONINDEXED -
           RECORDFORMAT(NOCIFORMAT) -
           SHR(3) -
           TRACKS(primary secondary)) - _____ 5
           DATA(NAME(CVSE.LIBRARY.DATA))
           "
/*           |_____ 6
/ &
```

**Note:** A hyphen at the end of a statement indicates a continuation of the VSAM parameter. The hyphen may be in any column but must be preceded by at least one blank.

Figure 1. Define the C/VSE Library Cluster in VSAM Space.

- 1 Specify the catalog name.

Change the catalog name (VSESP.USER.CATALOG in the example) to the catalog in which the C/VSE library is to be defined.

- 2 Delete the cluster before redefining.

An IDCAMS DELETE statement is included before the DEFINE statement so that the job can be rerun if necessary. When the job is run for the first time, the DELETE action will result in the following messages:

```
IDC3012I ENTRY CVSE.LIBRARY NOT FOUND
IDC3009I ** VSAM CATALOG RETURN CODE IS 8 - REASON CODE IS IGG0CLCG-6
IDC0551I **ENTRY CVSE.LIBRARY NOT DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 8
```

These messages can be ignored. IDCAMS will continue to define the cluster.

- 3 Specify the cluster name.

Change the cluster name (CVSE.LIBRARY in the example) of the C/VSE library to suit your installation.

- 4 Specify the volume serial number.

Change the *volser* parameter to the serial number of the volume on which the C/VSE library is to be defined.

- 5 Specify the amount of space.

Change the *primary* and *secondary* parameters to specify the number of tracks to be allocated for primary and secondary space for the C/VSE library. If you are using FBA devices, change this parameter to

```
BLOCKS(primary secondary)
```

For information about library storage requirements, see ["DASD Storage" in topic 1.1.2.2.](#)

- 6 Specify the data component name.

Change the data component name (CVSE.LIBRARY.DATA in the example) of the C/VSE library to conform with the cluster name you specified at [3](#).

When you have defined the VSAM cluster for the library, use the sample job in [Figure 2](#) to define the library and sublibrary. You will need to tailor some statements to suit your installation. The tailoring requirements for each statement are discussed in the notes following [Figure 2](#).

```
// JOB  EDCXILDF
*
*   Define the C/VSE Library
*
*   _____ 1
// DLBL  IJSYSUC,'VSESP.USER.CATALOG',,VSAM
*
*   _____ 2
// DLBL  CVSE,'CVSE.LIBRARY',,VSAM
// EXEC  LIBR _____ 3
DELETE LIB=CVSE
DEFINE LIB=CVSE
DEFINE SUBLIB=CVSE.BASE
/*      "
/ &    | _____ 4
```

---

Figure 2. Define the LE/VSE Library.

**1** Specify the VSAM catalog.

Change the catalog name (VSESP.USER.CATALOG in the example) to match the name of the catalog you specified at **1** in [Figure 1](#).

**2** Specify the label information.

Change the file-id (CVSE.LIBRARY in the example) of the C/VSE library to match the name you specified at **3** in [Figure 1](#).

**3** Invoke the Librarian to define the library.

The Librarian job step includes a LIBR DELETE statement before the LIBR DEFINE statement, so the job can be rerun if necessary. This means the following messages will be issued when the job runs for the first time:

```
L255I  LIBRARY CVSE - OPEN FAILURE
L027I  ABNORMAL END DURING DELETE COMMAND
        PROCESSING
L113I  RETURN CODE OF DELETE IS 8
```

These messages are informational and can be ignored. The job continues to allocate the library.

**4** Specify the sublibrary.

Change the sublibrary name to suit your installation.

**Note:** You can also use the Interactive Interface dialog "Resource Definition" to define a library in VSAM space.

---

## 2.3.2 Defining a Library in Non-VSAM Space

On the disk volume to be used for the library, identify suitable areas of free space. To do this, list the volume table of contents (VTOC) of the disk to be used. [Figure 3](#) shows a job to list the VTOC of the volume with serial number SYSWK1. Change the volume serial number in the ASSGN statement **1** as required.

---

```
// JOB  EDCXIVTC
*
*   List Volume Table of Contents
*
// ASSGN SYS004,DISK,TEMP,VOL=SYSWK1,SHR
// ASSGN SYS005,SYSLST          "
// EXEC LVTOC                   |_____ 1
/*
/ &
```

---

Figure 3. List the Contents of Volume SYSWK1.

When you have decided where to allocate the C/VSE library, you can use the sample job shown in [Figure 4](#) to help you define the library. You will need to tailor some statements to suit your installation. The tailoring requirements for each statement are discussed in the notes following [Figure 4](#).

---

```
// JOB  EDCXINDF
*
*   Define a Non-VSAM Library for C/VSE
*
// OPTION LOG
// DLBL  CVSE, 'CVSE.LIBRARY',99/365,SD _____ 1
// EXTENT SYS002,volser,,,rtrk,nrk _____ 2
// ASSGN SYS002,DISK,VOL=volser,SHR _____ 3
// EXEC  LIBR _____ 4
DELETE LIB=CVSE
DEFINE LIB=CVSE
DEFINE SUBLIB=CVSE.BASE
/*
/ & _____ 5
```

---

Figure 4. Define the LE/VSE Library.

1 Specify the label information.

Change the filename (CVSE in the example) and file-id (CVSE.LIBRARY in the example) of the C/VSE library to suit your installation.

2 Specify the extent information:

- The *volser* parameter indicates the volume serial number of the volume on which the C/VSE library is to be defined. Change this parameter as required.
- The *rtrk* parameter indicates the beginning track or FBA block of the disk extent. Change this parameter as required.
- The *ntrk* parameter indicates the number of tracks or FBA blocks to be allocated for the C/VSE library. Change this parameter as required. For information about library storage requirements, see ["DASD Storage" in topic 1.1.2.2](#).

3 Specify the volume serial number.

The *volser* parameter indicates the volume serial number of the volume on which the C/VSE library is to be defined. Change this parameter to match the volume serial number specified on the EXTENT statement.

4 Invoke the Librarian to define the library.

The Librarian job step includes a LIBR DELETE statement before the LIBR DEFINE statement, so the job can be rerun if necessary. This means the following messages will be issued when the job runs for the first time:

```
L101I  LIBRARY CVSE DOES NOT EXIST
L027I  ABNORMAL END DURING DELETE COMMAND
      PROCESSING
L113I  RETURN CODE OF DELETE IS 8
```

These messages are informational and can be ignored. The job continues to allocate the library.

5 Specify the sublibrary.

Change the sublibrary name to suit your installation.

---

## 2.4 Step 3: Install C/VSE

You can install C/VSE using either the VSE/ESA Interactive Interface or a batch installation job.

Subtopics:

- [2.4.1 Method 1: Install C/VSE Using the Interactive Interface](#)
  - [2.4.2 Method 2: Install C/VSE Using a Batch Job](#)
- 

## 2.4.1 Method 1: Install C/VSE Using the Interactive Interface

The VSE/ESA Interactive Interface enables you to use dialog requests to install C/VSE. For more information about installing licensed programs using the Interactive Interface, see *VSE/ESA Installation*.

To install C/VSE using the Interactive Interface, do the following:

1. Logon to the Interactive Interface as the system administrator.
2. Mount the C/VSE distribution tape on an available tape drive.
3. In the following menus, specify the items that appear following the ==> symbol.
  - a. **VSE/ESA FUNCTION SELECTION** menu:

```
==> 1 (Installation)
```

- b. **INSTALLATION** menu:

```
==> 1 (Install Programs - V2 Format)
```

**Note:** You can use option 1 for both a stacked distribution tape containing one or more optional licensed programs, and a non-stacked tape containing just C/VSE. Both are in Librarian V2 format.

- c. **INSTALL PROGRAMS - V2 FORMAT** menu:

```
==> 1 (Prepare for Installation)
```

- d. **PREPARE FOR INSTALLATION** menu:

```
==> cuu (address of drive with distribution tape)
```

- e. **JOB DISPOSITION** menu:

Make any changes required and press ENTER to submit the job.



The output listing from this job lists the optional programs on the distribution tape, with program identifiers and recommended library sizes. The program identifier for each C/VSE component has the format C/VSE.xxxx.1.1.0.

- f. Check the output from the batch job created by the previous steps to ensure that it was successful. Once the batch job created by these steps has successfully run, the program identifiers of the optional programs on the distribution tape are also automatically entered on the **INSTALL ADDITIONAL PROGRAM(S) FROM TAPE** menu.
- g. After the batch job created by the previous step has successfully run, return to the **VSE/ESA FUNCTION SELECTION** menu:

==> 1 (Installation)

- h. **INSTALLATION** menu:

==> 1 (Install Programs - V2 Format)

- i. **INSTALL PROGRAMS - V2 FORMAT** menu:

==> 2 (Install Program(s) from Tape)

- j. **INSTALL ADDITIONAL PROGRAM(S) FROM TAPE** menu:

Enter 1 (install) in the OPT field against the required identifiers of the format C/VSE.xxxx.1.1.0, and 2 (skip installation) against any other optional licensed programs you do not intend to install at this time.

If you are not installing C/VSE in the default sublibrary PRD2.DBASE, enter the name of your library and sublibrary on this screen.

- k. Press PF5 to generate the installation job.

- l. **INSTALL ADDITIONAL PROGRAM(S) FROM TAPE** menu:

==> 1 (Save the list)

- m. **INSTALL ADDITIONAL PROGRAM(S) FROM TAPE** menu:

==> cuu (address of drive with distribution tape)

- n. **JOB DISPOSITION** menu:

Make any changes required and press ENTER to submit the job and install C/VSE.

- o. Check the output from the batch job created by the previous steps to ensure that the install was successful. If you are installing C/VSE from a non-stacked tape, you receive the following message:

```
IESI0083I  TAPE IS NOT V2-STACKED
```

This message is for information only and can be ignored. The Librarian RESTORE job ends with a return code of 4. This is not an error. Installation was successful.

Subtopics:

- [2.4.1.1 Condition Code and Messages](#)
  - [2.4.1.2 List the Directories and Components](#)
- 

### 2.4.1.1 Condition Code and Messages

If you receive a condition code greater than 4:

- Check the list output for error conditions.
  - See *VSE/ESA Messages and Codes* for corrective action.
  - Correct the error.
  - Rerun the job.
  - Recheck the condition code.
- 

### 2.4.1.2 List the Directories and Components

If you want to print the directory entries from the sublibrary where C/VSE is installed, and the component records from the system history file, you can now submit a batch job to do this. The last two steps of the job shown in [Figure 5 in topic 2.4.2](#) will produce these lists.

---

## 2.4.2 Method 2: Install C/VSE Using a Batch Job

A sample batch job to install C/VSE using MSHP is shown in [Figure 5](#). This sample job uses the MSHP system history file that already exists as part of the VSE system. This system history file might already be defined in the system standard labels; if not, make sure that DLBL and EXTENT statements, with the necessary information for the system history file, are included in the job stream.

Depending on how you ordered the C/VSE licensed program, you will have received a stacked tape containing one or more optional licensed programs, or a non-stacked tape containing only the C/VSE licensed program. The job shown in [Figure 5](#) will handle both stacked and non-stacked tapes.

To install C/VSE using a batch job, create and tailor the job stream shown in [Figure 5](#), mount the distribution tape, and run the installation job.

You will need to tailor some statements to suit your installation. The

tailoring requirements for each statement are discussed in the notes following [Figure 5](#).

---

```

// JOB  EDCXINST
*
*   Install C/VSE
*
// OPTION LOG
* _____ 1
*
* // DLBL  CVSE,'CVSE.LIBRARY'
* // EXTENT SYS005
* // ASSGN  SYS005,DISK,VOL=volser,SHR
*
// ASSGN  SYS006,cpu _____ 2
*
// MTC    REW,SYS006
* -----
* This step installs C/VSE
* from the distribution tape
* using the VSE system history file
* -----
// EXEC  MSHP,SIZE=900K,PARM='PIDSTACKED' _____ 3
INSTALL PROD FROMTAPE ID='C/VSE.BASE.1.1.0' -
  PROD INTO=PRD2.DBASE
INSTALL PROD FROMTAPE ID='C/VSE.JPN..1.1.0' - _____ 4
  PROD INTO=PRD2.DBASE
/*
* -----
* List the C/VSE Library
* -----
// EXEC  LIBR,PARM='MSHP' _____ 5
LISTDIR SUBLIB=PRD2.DBASE -
  OUTPUT=NORMAL -
  UNIT=SYSLST
/*
* -----
* Retrace the C/VSE product
* -----
// EXEC  MSHP,SIZE=900K _____ 6
RETRACE COMPONENT IDENTIFIER=5686-A01-00
RETRACE COMPONENT IDENTIFIER=5686-A01-01
/*
// MTC  RUN,SYS006
/*
/&

```

---

Figure 5. Install C/VSE

1 Specify the label information.

If you are installing C/VSE into a sublibrary other than the default (PRD2.DBASE), code the DLBL job control statement (EXTENT and ASSGN statements if necessary) to match those you specified in [Figure 2 in topic 2.3.1](#) or [Figure 4 in topic 2.3.2](#). The library name must match the name used in the LIBR job.

Usually, you do not need to specify label information for the system history file. Your installation should have a permanent system standard label for this, with IJSYSHF as the filename. (IJSYSHF is the default filename that MSHP looks for in a label statement.)

2 Assign the distribution tape.

Replace *cuu* with the address of the tape drive on which you have mounted the distribution tape.

3 Install C/VSE.

This job step invokes MSHP to install C/VSE into the sublibrary identified on the INTO operand of the INSTALL statement. If you are installing C/VSE into a sublibrary other than the default, change the name of the sublibrary on the INTO operand of each INSTALL statement to the name of the sublibrary into which you are installing the component specified in the FROMTAPE ID operand. For example, if you are installing C/VSE into sublibrary CVSE.BASE, change PRD2.DBASE to CVSE.BASE.

For more information about the MSHP install options, see *VSE/ESA System Control Statements*.

4 Select national language feature components.

The C/VSE national language is the language used for the compile-time messages and compiler heading lines. The base component of C/VSE includes uppercase and mixed-case U.S. English national language support. You do not need to install mixed-case U.S. English as a separate component.

You can install the Japanese national language feature from the C/VSE distribution tape. If you do not want to install the Japanese national language feature, remove this INSTALL statement.

**Note:** If your default national language for LE/VSE is Japanese you **must** install the Japanese national language feature of C/VSE.

5 List the directory entries.

This job step invokes LIBR to list the directory entries of the sublibrary where C/VSE was installed. Remove this step if the directory lists are not required. If you are installing C/VSE into a sublibrary other than the default, change the name of the sublibrary in this job step accordingly.

The names of most modules in the C/VSE library have a three-character prefix of EDC.

6 Retrace the C/VSE product in the system history file.

This step prints the component records from the system history file for C/VSE. Remove this step if an MSHP retrace listing is not required.

If you need to rerun this install job, make sure you first restore the system history file, which you should have backed up before you started the install process. You should also rerun the library allocation job, if applicable.

## 2.5 Step 4: Verify the Installation of C/VSE

The sample job EDCXIVP, in the C/VSE sublibrary member EDCXIVP.Z, compiles, link-edits, and runs the verification program EDCXISMP.C. The installation verification program lets you check that your installation is successful by exercising representative features of C/VSE.

[Figure 6](#) shows the job EDCXIVP. You will need to tailor some statements to suit your installation. The tailoring requirements for each statement are discussed in the notes following [Figure 6](#).

**Note:** To successfully run the C/VSE compiler, or programs generated by C/VSE, LE/VSE and the LE/VSE C-specific base component must be installed.

```

* $$ JOB JNM=EDCXIVP,CLASS=A
* $$ PUN DISP=I,CLASS=A
// JOB EDCXIVP1
*
* C/370 Compiler Installation Validation
*
// ASSGN SYSIPT,YSRDR
// EXEC IESINSRT
// JOB EDCXIVP2
*
* _____ 1
*
// LIBDEF *,SEARCH=(lib.userlib,PRD2.DBASE,PRD2.SCEEBASE)
// EXEC EDCXISMP,SIZE=10K " "
*
* _____|_____ 2 |
* | _____|_____ 3
* $$ END
// LIBDEF *,SEARCH=(PRD2.DBASE,PRD2.SCEEBASE)
// LIBDEF PHASE,CATALOG=lib.userlib _____ 1
// OPTION CATAL,DECK,NOJCANCEL
// EXEC EDCCOMP,SIZE=EDCCOMP,PARM=' /SOURCE,TERM,NAME(EDCXISMP) '
#include "edcxismp.c"
/*
// EXEC LNKEDT
// ASSGN SYSIPT,YSRDR
// EXEC IESINSRT
// EXEC LISTLOG
#&
$ $$ EOJ
* $$ END
// EXEC LISTLOG
/ &
* $$ EOJ

```

Figure 6. Installation Verification

**Note:** The use of IESINSRT causes a second job (EDCXIVP2) to be punched back into the job queue for execution. The IESINSRT job as used here enables the DECK output produced by the compiler to be used as input to the second job EDCXIVP2. Two output listings will be returned for the installation verification job.

**1** Specify a user library.

Specify a user library and sublibrary where the verification phase EDCXISMP will be cataloged in step 1, and searched for in step 2.

**2** Specify the C/VSE sublibrary.

Specify the library and sublibrary where C/VSE was installed. If you installed C/VSE in the default library and sublibrary you do not need to change this.

**3** Specify the LE/VSE sublibrary.

Specify the library and sublibrary where LE/VSE was installed. If you installed LE/VSE in the default library and sublibrary you do not need to change this.

For successful installation verification the job control provided will create two jobs and return two output listings. The first job (EDCXIVP1):

- Compiles the C program and produces a return code of 0.
- Link-edits the compiled program into the user sublibrary specified in **1**. The link-edit produces a number of Weak Unresolved External References (WXTRNs), and a return code of 2. This is normal for the C/VSE compiler and does not indicate a problem.
- Punches a second job (EDCXIVP2) back into the VSE/POWER reader queue.

The second job EDCXIVP2 executes the C program. [Figure 7](#) shows the expected SYSLSST output from the second job (EDCXIVP2) of the installation verification.

---

```
// JOB EDCXIVP2
*
// LIBDEF *,SEARCH=(lib.userlib,PRD2.BASE,PRD2.SCEEBASE)
// EXEC EDCXISMP,SIZE=10K
```

```
Product Name: 5686-A01
Version 1 Release 1 Modification 00
Installation Date:(yy:ddd) 96:274
VALIDATION SUCCESSFUL
```

```
1S55I  LAST RETURN CODE WAS 0000
EOJ  EDCXIVP2  MAX.RETURN CODE=0000
```

---

Figure 7. Example Output from EDCXIVP2

**Note:** The installation date and the user library *lib.userlib* will vary in your SYSLST output.

---

## 3.0 Chapter 3. Customizing C/VSE

This chapter describes how to customize C/VSE compile-time options after installation.

Subtopics:

- [3.1 Customizing Compile-Time Option Defaults](#)
  - [3.2 Loading the Compiler into the Shared Virtual Area](#)
  - [3.3 Changing the National Language for C/VSE](#)
  - [3.4 Compile-Time Option Descriptions](#)
- 

### 3.1 Customizing Compile-Time Option Defaults

After installing C/VSE, you can modify the compile-time default options values.

To change the IBM-supplied compile-time options, you must:

- Modify and assemble the source that defines the default values for the compile-time options.
- Link-edit the C/VSE compiler phase EDCDP.

Subtopics:

- [3.1.1 Modify Default Values of Compile-Time Options](#)
  - [3.1.2 Link-Edit the C/VSE Compiler Phase EDCDP](#)
- 

#### 3.1.1 Modify Default Values of Compile-Time Options

The sample job, EDCXIOPT in [Figure 8](#), is supplied in member EDCXIOPT.Z of the C/VSE sublibrary, to help you modify and assemble the IBM-supplied

default C/VSE compile-time options.

The EDCZ110A macro in EDCXIOPT contains the options you can modify for your installation. ["Compile-Time Option Descriptions" in topic 3.4](#) describes the possible values for these options. For more detailed information on these options, see the [C/VSE User's Guide](#).

The sample job, as supplied, has the compile-time options set to the IBM-supplied default values. If you prefer different values to the default values, you can change them now.

When you have made the changes required, submit the job for assembly on your VSE system. This job assembles and catalogs a new options object module in the sublibrary you specify. Once this has run successfully, continue with the section ["Link-Edit the C/VSE Compiler Phase EDCDP" in topic 3.1.2](#). This completes the customization of the C/VSE default compile-time options.

You might need to tailor some statements to suit your installation. The tailoring requirements for each statement are discussed in the notes following [Figure 8](#).

```

* $$ JOB JNM=EDCXIOPT,CLASS=A
* $$ PUN DISP=I,CLASS=A
// JOB EDCXIOPT
*
* Assemble C/VSE Compiler Default Options
*
// ASSGN SYSIPT,SYSRDR
// EXEC IESINSRT
$ $$ LST DISP=D _____ 1
// JOB EDCXIOPT2
// EXEC LIBR,PARM='MSHP;ACCESS SUBLIB=PRD2.DBASE'
* $$ END
// LIBDEF *,SEARCH=(PRD2.DBASE)
// OPTION DECK " _____ 2
*
// EXEC ASMA90,SIZE=ASMA90
PUNCH 'CATALOG EDCZ110E.OBJ REPLACE=YES'
PRINT ON,GEN
OPTIONS EDCZ110A AGGR=0, *
CHECKOUT=0, *
CHKACCUR=1, *
CHKENUM=0, *
CHKEXTRN=0, *
CHKGENERAL=1, *
CHKGOTO=1, *
CHKINIT=1, *
CHKPARAM=1, *
CHKPORT=0, *
CHKPPCHK=1, *
CHKPPTRC=0, *
CHKTRUNC=1, *
CSECT=0, *
EXECOPS=1, *
EXPMAC=0, *
FLAG=0, *
GONUMB=0, *
HWOPTSTR=0, *
INLNON=0, *
INLNAUT=1, *
INLNREP=0, *

```



```

        INLNTHR=250,          *
        INLNLIM=1000,        *
        LANGLVL=4,          *
        LOCALE=0,           *
        LOCALE=,            *
        LONGNAME=0,         *
        MCOL1F=1,           *
        MCOL2F=72,          *
        MCOL1V=1,           *
        MCOL2V=32767,        *
        MEMORY=0,           *
        NESTINC=16,         *
        OFFSET=0,           *
        OPT=0,              *
        PPNLY=0,            *
        RENT=0,              *
        SONF=1,              *
        SONV=0,              *
        SCOL1F=73,          *
        SCOL2F=80,          *
        SCOL1V=0,           *
        SCOL2V=0,           *
        SHOWINC=0,          *
        SPILL=128,          *
        SSCOMM=0,           *
        TESTON=0,           *
        TESTBLK=1,          *
        TESTLIN=1,          *
        TESTPTH=0,          *
        TESTSYM=1,          *
        UPCONV=0            *
    END
/*
// ASSGN SYSIPT,SYSRDR
// EXEC IESINSRT
/*
#&
$ $$ EOJ
* $$ END
/ &
* $$ EOJ

```

Figure 8. Job to assemble and catalog the default compile-time options

**Note:** The use of IESINSRT causes a second job (EDCXIOP2) to be punched back into the job queue for execution. Job EDCXIOP2 will catalog the output from the assembly.

- 1 Specify the sublibrary.

Change PRD2.DBASE to the sublibrary where the modified C/VSE default options will be cataloged. If you do not want to replace the IBM-supplied default module, you might need to change this to a user library.

- 2 Specify the sublibrary where C/VSE was installed.

If C/VSE was installed in a sublibrary other than the installation default of PRD2.DBASE, change this to reflect the installation sublibrary.

Both steps of EDCXIOPT produce a return code of 0. Any other return code indicates an error has occurred and must be investigated.

### 3.1.2 Link-Edit the C/VSE Compiler Phase EDCDP

The sample job EDCXILNK, provided in member EDCXILNK.Z in the C/VSE sublibrary, link-edits the C/VSE compiler phase EDCDP. This phase must be link-edited to successfully change the default compile-time options.

You will need to tailor some job control statements to suit your installation. The tailoring requirements for each statement are discussed in the notes following [Figure 9](#). When this is completed, submit the job to your VSE system to link-edit the C/VSE compiler phase EDCDP.

```
// JOB  EDCXILNK
*
*   Link-edit the C/VSE compiler phase EDCDP
*
*   _____ 1
*   |           | 2
*
// LIBDEF *,SEARCH=(lib.userlib,PRD2.DBASE)
// LIBDEF PHASE,CATALOG=PRD2.DBASE
// OPTION CATAL      "
INCLUDE EDCOPLNK    |_____ 3
// EXEC  LNKEDT,PARM='MSHP'
/*
/ &
```

Figure 9. Job to link-edit the default options phase

- 1 Specify the first search sublibrary.

If your default options module was cataloged into the default library PRD2.DBASE (see **1** in [Figure 8](#)) when you assembled the changed compile-time options, then remove this user sublibrary. Otherwise replace the sublibrary *lib.userlib* with the name of the sublibrary where your customized default options module was cataloged in the section ["Modify Default Values of Compile-Time Options" in topic 3.1.1.](#)

- 2 Specify the second search sublibrary.

If C/VSE was installed in a sublibrary other than the installation default of PRD2.DBASE, change this to reflect your sublibrary. This sublibrary is where the object modules and the link book for the C/VSE compiler can be found.

- 3 Specify the sublibrary for the compiler.

If C/VSE was installed in a sublibrary other than the installation default of PRD2.DBASE, change this to reflect your sublibrary. This is where the new C/VSE compiler phase EDCDP will be cataloged. If you do not want to replace the IBM-supplied version of the compiler, you need to change this sublibrary, and then ensure this new sublibrary is always concatenated in the library search chain before the C/VSE compiler sublibrary when compiling programs.

The link-edit produces a number of Weak Unresolved External References (WXTRNs), and a return code of 2. This is normal for the link-edit of EDCDP and does not indicate a problem.

---

## 3.2 Loading the Compiler into the Shared Virtual Area

Placing routines in the SVA reduces overall system storage requirements. Also initialization and termination time is reduced for each application, since load time decreases.

All of the phases listed in [Table 12 in topic 1.2.3](#) can be included in the SVA. To include them:

1. Modify the SVA statement of the VSE IPL ASI (Automated System Initialization) procedure to allow space for the phases:
  - Increase the SDL parameter by the number of new phases being added to the SVA.
  - Increase the PSIZE parameter by the amount of storage required to contain the new phases being added to the 31-bit SVA.
2. Modify the VSE background (BG) ASI procedure to automatically load the selected phases into the SVA:
  - Modify the ALLOC statements for the partitions to ensure that the remaining SVA storage is large enough to contain the selected phases.
  - Modify the LIBDEF PHASE SEARCH job control statement preceding the SET SDL statement to include the name of the sublibrary containing the C/VSE phases.
  - After the SET SDL statement, add the statement:

```
LIST=$SVAEDCC
```

or, if not all phases are required to be loaded into the SVA, add the statement:

```
    phasename, SVA
```

for each phase required to be loaded into the SVA.

3. Shut down and re-IPL your VSE system.

The supplied SVA loadlist member \$SVAEDCC contains all the compiler modules that are eligible for including in the SVA. These are listed in [Table 12 in topic 1.2.3](#).

For more information on including phases in the SVA, see *VSE/ESA System Control Statements*.

### 3.3 Changing the National Language for C/VSE

The C/VSE compiler executes in a LE/VSE environment, therefore the compiler default national language is the same as the LE/VSE default for the national language. To change the LE/VSE default for the national language, see the *LE/VSE Installation and Customization Guide*.

To change the national language used for individual executions of the compiler use the PARM parameter of the job control EXEC statement, as shown in [Figure 10](#).

```
// EXEC  EDCCOMP,SIZE=EDCCOMP,PARM='NATLANG(xxx)'/
                                     |
                                     1 _____|
```

Figure 10. Example of the PARM option with NATLANG

- 1 Specify the national language required, where xxx may be:

```
ENU  Mixed-case U.S. English
UEN  Uppercase U.S. English
JPN  Japanese
```

### 3.4 Compile-Time Option Descriptions

[Table 11 in topic 1.2.2](#) lists the C/VSE compile-time options, their IBM-supplied default values and your planned customization.

C/VSE compile-time options are described on the following pages. For detailed descriptions of the option formats, see the [C/VSE User's Guide](#).

In the following syntax diagrams, unless otherwise stated:

- 0 Indicates the option is off. For example, EXECOPS=0 sets the default compile-time option to NOEXECOPS.
- 1 Indicates the option is on. For example, AGGR=1 sets the default compile-time option to AGGREGATE.

The options are listed here according to the names used in the customization macro (see [Figure 8 in topic 3.1.1](#)). Some of the names used in the macro do not match the actual compile-time option names. For example, the CHKACCR macro option corresponds to the CHECK(ACCURACY) compile-time option. Where necessary, the following descriptions identify the actual compile-time option name.

#### Subtopics:

- [3.4.1 AGGR](#)
- [3.4.2 CHECKOUT](#)
- [3.4.3 CHKACCUR](#)
- [3.4.4 CHKENUM](#)
- [3.4.5 CHKEXTRN](#)
- [3.4.6 CHKGENERAL](#)
- [3.4.7 CHKGOTO](#)
- [3.4.8 CHKINIT](#)
- [3.4.9 CHKPARM](#)
- [3.4.10 CHKPORT](#)
- [3.4.11 CHKPPCHK](#)
- [3.4.12 CHKPPTRC](#)
- [3.4.13 CHKTRUNC](#)
- [3.4.14 CSECT](#)
- [3.4.15 DECK](#)
- [3.4.16 EXECOPS](#)
- [3.4.17 EXPMAC](#)
- [3.4.18 FLAG](#)
- [3.4.19 GONUMB](#)
- [3.4.20 HWOPTSTR](#)
- [3.4.21 INFILE](#)
- [3.4.22 INLNON](#)
- [3.4.23 INLNAUT](#)
- [3.4.24 INLNREP](#)
- [3.4.25 INLNTHR](#)
- [3.4.26 INLNLIM](#)
- [3.4.27 LANGLVL](#)
- [3.4.28 LIST](#)
- [3.4.29 LOCALEON](#)
- [3.4.30 LOCALE](#)
- [3.4.31 LONGNAME](#)
- [3.4.32 LSEARCH](#)
- [3.4.33 MCOL1F](#)
- [3.4.34 MCOL2F](#)
- [3.4.35 MCOL1V](#)
- [3.4.36 MCOL2V](#)
- [3.4.37 MEMORY](#)

- [3.4.38 NAME](#)
  - [3.4.39 NESTINC](#)
  - [3.4.40 OBJECT](#)
  - [3.4.41 OFFSET](#)
  - [3.4.42 OPT](#)
  - [3.4.43 PPNLY](#)
  - [3.4.44 RENT](#)
  - [3.4.45 SEARCH](#)
  - [3.4.46 SONF](#)
  - [3.4.47 SONV](#)
  - [3.4.48 SCOL1F](#)
  - [3.4.49 SCOL2F](#)
  - [3.4.50 SCOL1V](#)
  - [3.4.51 SCOL2V](#)
  - [3.4.52 SHOWINC](#)
  - [3.4.53 SOURCE](#)
  - [3.4.54 SPILL](#)
  - [3.4.55 SSCOMM](#)
  - [3.4.56 START](#)
  - [3.4.57 TARGET](#)
  - [3.4.58 TERM](#)
  - [3.4.59 TESTON](#)
  - [3.4.60 TESTBLK](#)
  - [3.4.61 TESTLIN](#)
  - [3.4.62 TESTPTH](#)
  - [3.4.63 TESTSYM](#)
  - [3.4.64 UPCONV](#)
  - [3.4.65 XREF](#)
- 

### 3.4.1 AGGR

```
>>__AGGR=_0_|_1_|___><
```

Specifies whether the compiler generates a list of structures and unions and their sizes.

---

### 3.4.2 CHECKOUT

```
>>__CHECKOUT=_0_|_1_|___><
```

Specifies whether the compiler gives informational error messages indicating possible programming errors. This aids C programmers in debugging their programs.

You can specify CHECKOUT with or without suboptions. The following suboptions, CHKACCUR to CHKTRUNC, either enable or suppress the messages. Any of these suboptions may be given a default value, but the suboption is only meaningful when CHECKOUT is in effect.

---

### 3.4.3 CHKACCUR

\_1\_

```
>>__CHKACCUR=_|_0_|___><
```

Represents the ACCURACY suboption of CHECKOUT; enables or suppresses messages for assignments of long values to variables that are not long.

---

### 3.4.4 CHKENUM

```
>>__CHKENUM=_|_0_|___><
```

Represents the ENUM suboption of CHECKOUT; enables or suppresses messages for usage of enumerations.

---

### 3.4.5 CHKEXTRN

```
>>__CHKEXTRN=_|_0_|___><
```

Represents the EXTERN suboption of CHECKOUT; enables or suppresses messages for unused variables that have external declarations.

---

### 3.4.6 CHKGENERAL

```
>>__CHKGENERAL=_|_1_|___><
```

Represents the GENERAL suboption of CHECKOUT; enables or suppresses general checkout messages.

---

### 3.4.7 CHKGOTO

```
>>__CHKGOTO=_|_1_|___><
```

Represents the GOTO suboption of CHECKOUT; enables or suppresses messages for the appearance and usage of goto statements.

---

### 3.4.8 CHKINIT

```
>>__CHKINIT=_|_1_|___><
```

Represents the INIT suboption of CHECKOUT; enables or suppresses messages for variables that are not explicitly initialized.

---

### 3.4.9 CHKPARM

```
>>__CHKPARM=_  1  |_  0  |___><
```

Represents the PARM suboption of CHECKOUT; enables or suppresses messages for function parameters that are not used.

---

### 3.4.10 CHKPORT

```
>>__CHKPORT=_  0  |_  1  |___><
```

Represents the PORT suboption of CHECKOUT; enables or suppresses messages for non-portable use of the C language.

---

### 3.4.11 CHKPPCHK

```
>>__CHKPPCHK=_  1  |_  0  |___><
```

Represents the PPCHECK suboption of CHECKOUT; enables or suppresses messages for all preprocessor directives.

---

### 3.4.12 CHKPPTRC

```
>>__CHKPPTRC=_  0  |_  1  |___><
```

Represents the PPTRACE suboption of CHECKOUT; enables or suppresses messages for the tracing of include files by the preprocessor.

---

### 3.4.13 CHKTRUNC

```
>>__CHKTRUNC=_  1  |_  0  |___><
```

Represents the TRUNC suboption of CHECKOUT; enables or suppresses messages for variable names that are truncated by the compiler.

---



### 3.4.14 CSECT

```
>>__CSECT=|_0_|_1_|___><
```

Specifies whether the compiler generates error messages if your source program does not contain the named CSECTs.

---

### 3.4.15 DECK

Specifies the compiler is to write an object module to SYSPCH. This option can only be specified using the // OPTION DECK job control statement and is documented here for reference only.

---

### 3.4.16 EXECOPS

```
>>__EXECOPS=|_0_|_1_|___><
```

Specifies whether the program accepts run-time options from the command line.

---

### 3.4.17 EXPMAC

```
>>__EXPMAC=|_0_|_1_|___><
```

Specifies whether the compiler shows all expanded macros in the source listing. To use this option, you must also use the SOURCE option to generate a source listing.

---

### 3.4.18 FLAG

```
>>__FLAG=|_0_|_10_|_30_|_40_|___><
```

Specifies the minimum severity of error for which the compiler generates a message:

- 0 List all messages. Equivalent to FLAG(I) or NOFLAG.
- 10 List all except information messages. Equivalent to FLAG(W).
- 30 List all except warning and information messages. Equivalent to

- FLAG(E).  
40 List only severe error and unrecoverable error messages. Equivalent to FLAG(S).
- 

### 3.4.19 GONUMB

```
>>__GONUMB=_0_|_1_|___><
```

Specifies whether the compiler generates line number tables corresponding to the input source file. These tables are used by Debug Tool for VSE/ESA, and for error tracebacks.

---

### 3.4.20 HWOPTSTR

```
>>__HWOPTSTR=_0_|_1_|___><
```

Specifies whether the compiler generates code to take advantage of different hardware.

---

### 3.4.21 INFILE

Specifies the compiler is to take its source input from a file rather than SYSIPT. This option can only be specified in the PARM parameter of the // EXEC job control statement, and is documented here for reference only.

---

### 3.4.22 INLNON

```
>>__INLNON=_0_|_1_|___><
```

Specifies whether the compiler is to place the code for the function at the point of call. The INLINE compile-time option has four optional parameters represented by suboptions, INLNAUT to INLNLIM, following. Any of these suboptions may be given a default value, but the suboption is only meaningful when INLINE is in effect.

---

### 3.4.23 INLNAUT

```
>>__INLNAUT=_1_|_0_|___><
```

Represents the AUTO suboption of INLINE; specifies whether the compiler

runs in automatic mode and inlines functions within the *threshold* (specified by the INLNTHR suboption) and *limit* (specified by the INLNLIM suboption).

---

### 3.4.24 INLNREP

```
>>__INLNREP=|_0_|____<<
```

Represents the REPORT suboption of INLINE; specifies an inline report is part of the compiler listing file, consisting of:

- Inline summary
  - Detailed call structure
- 

### 3.4.25 INLNTHR

```
>>__INLNTHR=|_250_|____<<
```

Represents the *threshold* suboption of INLINE; specifies the maximum relative size of a function to inline.

---

### 3.4.26 INLNLIM

```
>>__INLNLIM=|_1000_|____<<
```

Represents the *limit* suboption of INLINE; specifies the maximum relative size to which a function can grow before auto-inlining stops.

---

### 3.4.27 LANGLVL

```
>>__LANGLVL=|_4_|____<<
              |_1_|
              |_2_|
              |_3_|
```

Specifies the language standard for the compiler.

- 1 Only those language constructs that conform to ANSI C standards are available for use in your C program. Equivalent to LANGLVL(ANSI).
- 2 Only those language constructs that conform to SAA Level 1 standards are available for use in your C program. Equivalent to LANGLVL(SAA).
- 3 Only those language constructs that conform to SAA Level 2 standards are available for use in your C program. Equivalent to LANGLVL(SAAL2).

- 4 All IBM system/370 extensions to the C language are available for use in your C program. Equivalent to LANGLVL(EXTENDED).
- 

### 3.4.28 LIST

Determines whether the compiler generates a pseudo assembler listing. The default value for this option is taken from the VSE system option LISTX which can be specified by the // OPTION LISTX job control statement, or in the system defaults at IPL time. The option is documented here for reference only.

---

### 3.4.29 LOCALEON

```
>>__LOCALEON=_  0  ____<<
          |_1_|
```

Indicates that the default current locale, or the locale named in the LOCALE option, is to be used by the compiler as the current locale throughout the compilation.

---

### 3.4.30 LOCALE

```
>>__LOCALE=_          ____<<
          |_name_|
```

Specifies the name of the locale to be used by the compiler when LOCALEON=1. If no name is specified, the default current locale in the environment is used.

---

### 3.4.31 LONGNAME

```
>>__LONGNAME=_  0  ____<<
          |_1_|
```

Specifies that the compiler generates long and mixed-case names in the object module produced by the compiler.

---

### 3.4.32 LSEARCH

Determines in which library the preprocessor begins searching for the user include files. This option can only be specified when invoking the compiler, and is documented here for reference only.

---

### 3.4.33 MCOL1F

```
>>__MCOL1F= 1 | left-margin | ____><
```

Represents the first parameter of the MARGINS compile-time option for fixed-length files; specifies the left margin in the source file to be scanned for input to the compiler. The value of *left-margin* must be greater than 0 and less than 32768. This suboption is only meaningful for fixed-length input files.

---

### 3.4.34 MCOL2F

```
>>__MCOL2F= 72 | right-margin | ____><
```

Represents the second parameter of the MARGINS compile-time option for fixed-length files; specifies the right margin in the source file to be scanned for input to the compiler. The value of *right-margin* must be greater than MCOL1F and less than 32768. This suboption is only meaningful for fixed-length input files.

**Note:** To specify NOMARGINS for fixed-length files assign 1 to MCOL1F and 32767 to MCOL2F.

---

### 3.4.35 MCOL1V

```
>>__MCOL1V= 1 | left-margin | ____><
```

Represents the first parameter of the MARGINS compile-time option for variable-length files; specifies the left margin in the source file to be scanned for input to the compiler. The value of *left-margin* must be greater than 0 and less than 32768. This suboption is only meaningful for variable-length input files.

---

### 3.4.36 MCOL2V

```
>>__MCOL2V= 32767 | right-margin | ____><
```

Represents the second parameter of the MARGINS compile-time option for variable-length files; specifies the right margin in the source file to be scanned for input to the compiler. The value of *right-margin* must be greater than MCOL1V and less than 32768. This suboption is only meaningful for variable-length input files.

**Note:** To specify NOMARGINS for variable-length files assign 1 to MCOL1V

and 32767 to MCOL2V.

---

### 3.4.37 MEMORY

```
>>__MEMORY=_|_1_|___><
```

Determines whether the preprocessor and/or compiler uses MEMORY files, if possible, when a workfile is created. Using this option decreases compilation time, but might require extra memory. Memory files are allocated from C heap storage.

---

### 3.4.38 NAME

Specifies whether the compiler generates a control statement in front of the object module. This option can only be specified when invoking the compiler, and is documented here for reference only. The following are control statements generated for each job control OPTION, when the NAME option is set.

<b>DECK</b>	Generates a Librarian CATALOG statement
<b>LINK</b>	Generates a linkage editor PHASE statement
<b>CATAL</b>	Generates a linkage editor PHASE statement

---

### 3.4.39 NESTINC

```
>>__NESTINC=_|_n_|___><
```

Specifies the limit for the number of nested include files. The value of n must be in the range 0 to 32767

---

### 3.4.40 OBJECT

Specifies the compiler is to write the object module to SYSLNK. This option can only be specified using the // OPTION LINK or // OPTION CATAL job control statements, and is documented here for reference only.

---

### 3.4.41 OFFSET

```
>>__OFFSET=_|_1_|___><
```

Specifies whether the compiler displays the offset addresses of each

function in the object module listing. To use the OFFSET option, you must also use the LIST option to generate the object module listing.

---

### 3.4.42 OPT

```
>>__OPT=_  0  |_  1  |___><
```

Specifies whether the compiler optimizes the machine instructions to produce more efficient object code. Optimization can reduce the amount of main storage required for the object module. Using OPTIMIZE increases compile time.

- 0 Indicates no optimization is done.
  - 1 Indicates local optimizations are performed.
- 

### 3.4.43 PPOONLY

```
>>__PPOONLY=_  0  |_  1  |___><
```

Specifies whether only the preprocessor runs against the source file, as opposed to both the preprocessor and the compiler.

- 0 Both the preprocessor and the compiler run against the source file.
  - 1 Only the preprocessor runs against the source file.
- 

### 3.4.44 RENT

```
>>__RENT=_  0  |_  1  |___><
```

Specifies whether the compiler generates reentrant code.

---

### 3.4.45 SEARCH

Specifies in which library the preprocessor begins searching for the system include files. This option can only be specified when invoking the compiler, and is documented here for reference only.

---

### 3.4.46 SONF

  1

```
>>__SONF=_|_0_|___><
```

Specifies whether the fixed-length input records are to contain sequence numbers.

---

### 3.4.47 SONV

```
>>__SONV=_|_1_|___><
```

Specifies whether the variable-length input records are to contain sequence numbers.

---

### 3.4.48 SCOL1F

```
>>__SCOL1F=_|_73_|___><
```

Represents the first parameter of the SEQUENCE option for fixed-length input files and specifies the left margin of the sequence number. The value of *left-margin* must be greater than 0 and less than 32768. This suboption is only effective when SONF=1.

---

### 3.4.49 SCOL2F

```
>>__SCOL2F=_|_80_|___><
```

Represents the second parameter of the SEQUENCE option for fixed-length input files and specifies the right margin of the sequence number. The value of *right-margin* must be greater than SCOL1F and less than 32768. This suboption is only effective when SONF=1.

---

### 3.4.50 SCOL1V

```
>>__SCOL1V=_|_0_|___><
```

Represents the first parameter of the SEQUENCE option for variable-length input files and specifies the left margin of the sequence number. The value of *left-margin* must be greater than 0 and less than 32768. This suboption is only effective when SONV=1, and may be given a value of 0 when SONV=0.

---

### 3.4.51 SCOL2V



```
>>__SCOL2V=_  0  |_right-margin_|___><
```

Represents the second parameter of the SEQUENCE option for variable-length input files and specifies the right margin of the sequence number. The value of *right-margin* must be greater than SCOL1V and less than 32768. This suboption is only effective when SONV=1, and may be given a value of 0 when SONV=0.

---

### 3.4.52 SHOWINC

```
>>__SHOWINC=_  0  |_1_|___><
```

Specifies whether the compiler shows, in the compiler listing, all include files processed. To use the SHOWINC option, you must also use the SOURCE option.

---

### 3.4.53 SOURCE

Specifies whether the compiler shows, in the compiler listing, the source input statements. The default value for this option is taken from the VSE system option LIST which can be specified by the // OPTION LIST job control statement, or in the system defaults at IPL time. The option is documented here for reference only.

---

### 3.4.54 SPILL

```
>>__SPILL=_ 128 |_n_|___><
```

Specifies the size of the spill area to be used for the compilation. The value of n must be in the range 0 to 3900.

---

### 3.4.55 SSCOMM

```
>>__SSCOMM=_  0  |_1_|___><
```

Specifies that the compiler is to recognize two slashes (//) as the beginning of a comment that will terminate at the end of the line.

---

### 3.4.56 START

Specifies whether CEESTART is generated. In C/VSE CEESTART is always generated. The option is documented here for reference only.

---

### 3.4.57 TARGET

Specifies that the compiler generates object code for VSE/ESA systems. The option is documented here for reference only.

---

### 3.4.58 TERM

Specifies that all error messages from the compiler are to be directed to SYSLOG. The default value for this option is taken from the VSE system option TERM which can be specified by the // OPTION TERM job control statement, or in the system defaults at IPL time. The option is documented here for reference only.

---

### 3.4.59 TESTON

```
>>__TESTON=|_0_|____<<
```

Represents the TEST compile-time option and specifies whether the compiler generates information for debugging. When the TEST option is used then suboptions, TESTBLK to TESTSYM, apply. Any of these suboptions may be given a default value but the suboption is only meaningful when TEST is in effect.

---

### 3.4.60 TESTBLK

```
>>__TESTBLK=|_1_|____<<
```

Represents the BLOCK suboption of TEST, and specifies whether the compiler generates symbol information for nested blocks.

---

### 3.4.61 TESTLIN

```
>>__TESTLIN=|_1_|____<<
```

Represents the LINE suboption of TEST, and specifies whether the compiler generates line number hooks.

---

### 3.4.62 TESTPTH

```
>>__TESTPTH=|_0_|_1_|___><
```

Represents the PATH suboption of TEST, and specifies whether the compiler generates path break points.

---

### 3.4.63 TESTSYM

```
>>__TESTSYM=|_1_|_0_|___><
```

Represents the SYM suboption of TEST, and specifies whether the compiler generates symbol table information.

---

### 3.4.64 UPCONV

```
>>__UPCONV=|_0_|_1_|___><
```

Specifies whether the compiler follows *unsignedness preserving* rules when doing C type conversions.

---

### 3.4.65 XREF

Specifies whether the compiler includes in the compiler listing a cross reference table of names used in the program, together with the numbers of the lines where they are declared or referenced. The default value for this option is taken from the VSE system option XREF which can be specified by the // OPTION XREF job control statement, or in the system defaults at IPL time. The option is documented here for reference only.

---

## 4.0 Chapter 4. Maintaining C/VSE

This chapter describes how to replace or re-install C/VSE and how to apply service updates to C/VSE. To effectively use the maintenance procedures, you must have already installed C/VSE and any required products.

In addition, this chapter describes how to remove C/VSE.

Subtopics:

- [4.1 Re-Installing C/VSE](#)
  - [4.2 Applying Service Updates](#)
  - [4.3 Removing C/VSE](#)
  - [4.4 How to Report a Problem with C/VSE](#)
- 

## 4.1 Re-Installing C/VSE

You do not need to remove C/VSE from your system before re-installing C/VSE. However, if you intend to re-install the product in a different sublibrary from the previous installation, you must remove C/VSE from the system history file before you can re-install it. If you are re-installing in the same sublibrary, you might need to delete C/VSE from the sublibrary and release the space to ensure there is sufficient library space available for the re-installation.

If you do need to remove C/VSE from your system, the section ["Removing C/VSE" in topic 4.3](#) describes how to do this.

To re-install C/VSE, follow the same steps as for installing C/VSE. See [Chapter 2, "Installing C/VSE" in topic 2.0](#).

---

## 4.2 Applying Service Updates

You might need to apply maintenance or service updates to C/VSE periodically. There are two types of formally supported software fixes. One is the program temporary fix (PTF) applied as corrective service or as preventive maintenance. The other is the authorized program analysis report (APAR) fix applied as a code replacement in a corrective maintenance mode.

Subtopics:

- [4.2.1 What You Receive](#)
  - [4.2.2 Checklist for Applying Service](#)
  - [4.2.3 Step 1. Check Prerequisite APARs or PTFs](#)
  - [4.2.4 Step 2. Backup Original System](#)
  - [4.2.5 Step 3. Apply Service](#)
  - [4.2.6 Step 4. Run the Installation Verification Program \(IVP\)](#)
- 

### 4.2.1 What You Receive

If you report a problem with C/VSE to your IBM Support Center, you will receive a tape containing one or more APARs or PTFs that have been created to solve your problem.

You might also receive a list of prerequisite APARs or PTFs that should have been applied to your system before applying the current service. These prerequisite APARs or PTFs might relate to C/VSE or any other licensed product you have installed, including VSE/ESA.

You apply service to C/VSE using either the VSE/ESA Interactive Interface or a batch job.

The following checklist provides a summary of steps you should use to apply service to C/VSE.

## 4.2.2 Checklist for Applying Service

[Table 14](#) lists the steps for installing corrective service on C/VSE. You can use [Table 14](#) as a checklist.

Step	Description	Jobname	Topic
__ 1	Ensure prerequisite APARs or PTFs are applied	EDCXIRET	<a href="#">4.2.3</a>
__ 2	Backup existing system	___	<a href="#">4.2.4</a>
__ 3	Apply service	EDCXISER	<a href="#">4.2.5</a>
__ 4	Run the installation verification program	EDCXIVP	<a href="#">2.5</a> , <a href="#">4.2.6</a>

## 4.2.3 Step 1. Check Prerequisite APARs or PTFs

Prerequisite APARs or PTFs are APARs or PTFs that need to be applied to your system before you can apply the current maintenance. These APARs or PTFs might apply to C/VSE or any licensed program you have installed at your installation.

Your IBM Support Center will have given you a list of any relevant prerequisite APARs or PTFs. Most probably they will already be applied to your system. You can verify this by retracing the APARs and PTFs in your system history file. The job shown in [Figure 11](#) shows how to retrace APARs and PTFs in the system history file.

```
// JOB EDCXIRET
*
*   Retrace APARs and PTFs
*
// EXEC MSHP,SIZE=900K
RETRACE APARS
RETRACE PTFS
/*
/ &
```

---

Figure 11. Retrace APARs and PTFs

Use the listing produced by this job to check whether you have applied the prerequisite APARs or PTFs. If you have not, your IBM Support Center will arrange to send them to you and you should apply them before applying other service.

---

## 4.2.4 Step 2. Backup Original System

Make a backup copy of your current C/VSE library and the system history file. For information about backing up libraries and the system history file, see *VSE/ESA System Control Statements*.

---

## 4.2.5 Step 3. Apply Service

You can apply service to C/VSE from the provided service tape using either the Interactive Interface or a batch job.

You will receive detailed instructions for applying service with the service tape.

Subtopics:

- [4.2.5.1 Method 1: Apply Service Using the Interactive Interface](#)
  - [4.2.5.2 Method 2: Apply Service Using a Batch Job](#)
- 

### 4.2.5.1 Method 1: Apply Service Using the Interactive Interface

The VSE/ESA Interactive Interface enables you to use dialog requests to apply service to C/VSE. For more information about the functions of the Interactive Interface, see *VSE/ESA Administration*. For more information on how to use the dialogs to apply service, see *VSE/ESA System Upgrade and Service*.

---

### 4.2.5.2 Method 2: Apply Service Using a Batch Job

The batch job to apply service to C/VSE uses the MSHP system history file where C/VSE was installed.

A sample job to apply service using the Maintain System History Program (MSHP) is shown in [Figure 12](#). For more information on MSHP see *VSE/ESA*

*System Control Statements.*

---

```
// JOB  EDCXISER
*
*   Apply Service
*
// ASSGN SYS006,cuu _____ 1
// EXEC  MSHP,SIZE=900K
INSTALL SERVICE FROMTAPE _____ 2
/*
/ &
```

---

Figure 12. Apply Service

- 1 Specify the tape address.

Change cuu to the address of the tape drive where you have mounted the service tape.

- 2 This shows the MSHP statement to install service from a tape. The information in the system history file will direct MSHP to apply the service to the sublibrary in which C/VSE is installed. You do not need to supply this information.
- 

#### 4.2.6 Step 4. Run the Installation Verification Program (IVP)

After you have applied all the files on the service tape, run the installation verification program EDCXIVP to ensure that C/VSE functions properly. See the description of EDCXIVP in [Figure 6 in topic 2.5](#).

---

### 4.3 Removing C/VSE

You do not have to remove C/VSE from your system before installing a new version or release.

If you do have to remove C/VSE from your system for any reason, you must delete all the C/VSE entries from your sublibrary and remove C/VSE from the system history file. [Figure 14](#) shows the job control needed to remove C/VSE from the system history file.

To delete all C/VSE entries from your sublibrary, use the DELETE command of the LIBR program. The job shown in [Figure 13](#) shows the job control needed to delete C/VSE from the default sublibrary PRD2.DBASE. You might need to tailor some statements to suit your installation. The tailoring requirements for each statement are discussed in the notes following

[Figure 13.](#)

---

```
// JOB  EDCXIDEL
*
*   Delete C/VSE
*
* Label for the C/VSE library _____ 1
*
// EXEC  LIBR,SIZE=200K
ACCESS S=PRD2.DBASE _____ 2
DELETE EDC*.*
DELETE HDA01*.Z
DELETE $SVAEDCC.*
DELETE $SV$EDCC.*
/*
/ &
```

---

Figure 13. Delete C/VSE from a Sublibrary

**1** Specify the C/VSE library.

If you have installed C/VSE into a sublibrary other than the default (PRD2.DBASE) then insert the required DLBL, EXTENT, and ASSGN information for the C/VSE library.

**2** Specify the C/VSE library

If you have installed C/VSE in a sublibrary other than the default, change this statement.

To remove C/VSE from the system history file, use the REMOVE command of the Maintain System History Program (MSHP). The sample job shown in [Figure 14](#) shows the job control needed to remove C/VSE from the system history file.

---

```
// JOB  EDCXIREM
*
*   Remove the C/VSE product
*
// EXEC  MSHP,SIZE=900K
REMOVE 5686-A01-00-1FY
REMOVE 5686-A01-01-1G4 _____ 1
/*
/ &
```

---



Figure 14. Remove C/VSE from the System History File

1 Japanese national Language feature

Remove this control statement if you did not install the Japanese national language feature.

---

## 4.4 How to Report a Problem with C/VSE

For information on how to report a problem with C/VSE, see the [C/VSE Diagnosis Guide](#).

---

# BIBLIOGRAPHY Bibliography

Subtopics:

- [BIBLIOGRAPHY.1 IBM C for VSE/ESA Publications](#)
  - [BIBLIOGRAPHY.2 IBM Language Environment for VSE/ESA Publications](#)
  - [BIBLIOGRAPHY.3 Related Publications](#)
  - [BIBLIOGRAPHY.4 Softcopy Publications](#)
- 

## BIBLIOGRAPHY.1 IBM C for VSE/ESA Publications

*Licensed Program Specifications*, GC09-2421

*Installation and Customization Guide*, GC09-2422

[Migration Guide](#), SC09-2423

[User's Guide](#), SC09-2424

[Language Reference](#), SC09-2425

[Diagnosis Guide](#), GC09-2426

---

## BIBLIOGRAPHY.2 IBM Language Environment for VSE/ESA Publications

*Fact Sheet*, GC33-6679

*Concepts Guide*, GC33-6680

*Debugging Guide and Run-Time Messages*, SC33-6681

*Installation and Customization Guide*, SC33-6682

*Licensed Program Specifications*, GC33-6683

*Programming Guide*, SC33-6684

*Programming Reference*, SC33-6685

*Run-Time Migration Guide*, SC33-6687

*Writing Interlanguage Communication Applications*, SC33-6686

*C Run-Time Programming Guide*, SC33-6688

*C Run-Time Library Reference*, SC33-6689

---

## **BIBLIOGRAPHY.3 Related Publications**

### **VSE/ESA Version 1 Release 4**

*Planning*, SC33-6503

*Installation and Service*, SC33-6504

*Administration*, SC33-6505

*Messages and Codes*, SC33-6507

*Guide to System Functions*, SC33-6511

*System Control Statements*, SC33-6513

*VSE/VSAM Commands and Macros*, SC33-6535

*VSE/VSAM User's Guide*, SC33-6535

### **VSE/ESA Version 2**

*System Upgrade and Service*, SC33-6602

*Planning*, SC33-6603

*Installation*, SC33-6604

*Administration*, SC33-6605

*Messages and Codes*, SC33-6607

*Guide to System Functions*, SC33-6611

*System Control Statements*, SC33-6613

*VSE/VSAM Commands and Macros*, SC33-6631

*VSE/VSAM User's Guide and Application Programming*, SC33-6632

---

## BIBLIOGRAPHY.4 Softcopy Publications

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

*VSE Collection*, SK2T-0060

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

---

## INDEX    Index

# Special Characters

---

\$SVAEDCC loadlist, [1.2.3](#)  
[3.2](#)

## A

---

ACCURACY suboption, [3.4.3](#)  
AGGR option, [1.2.2](#)  
[3.4.1](#)  
allocating C/VSE library space, [2.3](#)  
APARs, [4.2.3](#)

applying service updates  
  backup original system, [4.2.4](#)  
  retracing APARs and PTFs, [4.2.3](#)  
  system history file, [4.2.3](#)  
  using a batch job, [4.2.5.2](#)  
  using the interactive interface, [4.2.5.1](#)  
  what you receive, [4.2.1](#)  
Assembler, High Level  
  minimum release, [1.1.2.1](#)  
AUTO suboption, [3.4.23](#)

## B

---

BLOCK suboption, [3.4.60](#)  
BookManager/Read  
  minimum release, [1.1.2.1](#)

## C

---

C-specific base component, [1.1.2.1](#)  
  [2.5](#)  
C/VSE  
  default library, [1.1.3](#)  
  library requirements, [1.1.2.2](#)  
  publications, [1.1.1.2](#)  
CATAL  
  VSE system option, [3.4.38](#)  
  [3.4.40](#)  
CATALOG statement  
  automatic generation of, [3.4.38](#)  
CEESTART, [3.4.56](#)  
CHECKOUT option, [1.2.2](#)  
  [3.4.2](#)  
CHKACCUR option, [1.2.2](#)  
  [3.4.3](#)  
CHKENUM option, [1.2.2](#)  
  [3.4.4](#)  
CHKEXTRN option, [1.2.2](#)  
  [3.4.5](#)  
CHKGENERAL option, [1.2.2](#)  
  [3.4.6](#)  
CHKGOTO option, [1.2.2](#)  
  [3.4.7](#)  
CHKINIT option, [1.2.2](#)  
  [3.4.8](#)  
CHKPARAM option, [1.2.2](#)  
  [3.4.9](#)  
CHKPORT option, [1.2.2](#)  
  [3.4.10](#)  
CHKPPCHK option, [1.2.2](#)  
  [3.4.11](#)  
CHKPPTRC option, [1.2.2](#)  
  [3.4.12](#)  
CHKTRUNC option, [1.2.2](#)  
  [3.4.13](#)  
CICS/VSE  
  minimum release, [1.1.2.1](#)  
CLC, [1.1.1](#)  
COBOL for VSE/ESA  
  minimum release, [1.1.2.1](#)  
compile-time options

ACCURACY suboption, [3.4.3](#)  
AGGR, [1.2.2](#)  
[3.4.1](#)  
AUTO suboption, [3.4.23](#)  
BLOCK suboption, [3.4.60](#)  
changing IBM-supplied defaults, [3.1](#)  
CHECKOUT, [1.2.2](#)  
[3.4.2](#)  
CHKACCUR, [1.2.2](#)  
[3.4.3](#)  
CHKENUM, [1.2.2](#)  
[3.4.4](#)  
CHKESTRN, [1.2.2](#)  
[3.4.5](#)  
CHKGENERAL, [1.2.2](#)  
[3.4.6](#)  
CHKGOTO, [1.2.2](#)  
[3.4.7](#)  
CHKINIT, [1.2.2](#)  
[3.4.8](#)  
CHKPARG, [1.2.2](#)  
[3.4.9](#)  
CHKPORT, [1.2.2](#)  
[3.4.10](#)  
CHKPPCHK, [1.2.2](#)  
[3.4.11](#)  
CHKPPTRC, [1.2.2](#)  
[3.4.12](#)  
CHKTRUNC, [1.2.2](#)  
[3.4.13](#)  
CSECT, [1.2.2](#)  
[3.4.14](#)  
customized installation defaults, [1.2.2](#)  
DECK, [3.4.15](#)  
ENUM suboption, [3.4.4](#)  
EXECOPS, [1.2.2](#)  
[3.4.16](#)  
EXPMAC, [1.2.2](#)  
[3.4.17](#)  
EXTERN suboption, [3.4.5](#)  
FLAG, [1.2.2](#)  
[3.4.18](#)  
GENERAL suboption, [3.4.6](#)  
GONUMB, [1.2.2](#)  
[3.4.19](#)  
GOTO suboption, [3.4.7](#)  
HWOPTSTR, [1.2.2](#)  
[3.4.20](#)  
IBM-supplied defaults, [1.2.2](#)  
[3.4](#)  
INFILE, [3.4.21](#)  
INIT suboption, [3.4.8](#)  
INLINE, [3.4.22](#)  
INLINE limit, [3.4.26](#)  
INLINE threshold, [3.4.25](#)  
INLNAUT, [1.2.2](#)  
[3.4.23](#)  
INLNLIM, [1.2.2](#)  
[3.4.26](#)  
INLNON, [1.2.2](#)  
[3.4.22](#)  
INLNREP, [1.2.2](#)  
[3.4.24](#)  
INLNTHR, [1.2.2](#)  
[3.4.25](#)  
LANGLVL, [1.2.2](#)  
[3.4.27](#)  
LINE suboption, [3.4.61](#)  
LIST, [3.4.28](#)  
LOCALE, [1.2.2](#)  
[3.4.30](#)

LOCALEON, [1.2.2](#)  
[3.4.29](#)

LONGNAME, [1.2.2](#)  
[3.4.31](#)

LSEARCH, [3.4.32](#)

MARGINS, [3.4.33](#)  
[3.4.34](#)  
[3.4.35](#)  
[3.4.36](#)

MCOL1F, [1.2.2](#)  
[3.4.33](#)

MCOL1V, [1.2.2](#)  
[3.4.35](#)

MCOL2F, [1.2.2](#)  
[3.4.34](#)

MCOL2V, [1.2.2](#)  
[3.4.36](#)

MEMORY, [1.2.2](#)  
[3.4.37](#)

NAME, [3.4.38](#)

NESTINC, [1.2.2](#)  
[3.4.39](#)

OBJECT, [3.4.40](#)

OFFSET, [1.2.2](#)  
[3.4.41](#)

OPT, [1.2.2](#)  
[3.4.42](#)

OPTIMIZE, [3.4.42](#)

PARM suboption, [3.4.9](#)

PATH suboption, [3.4.62](#)

planning to customize, [1.2.2](#)

PORT suboption, [3.4.10](#)

PPCHECK suboption, [3.4.11](#)

PPONLY, [1.2.2](#)  
[3.4.43](#)

PPTRACE suboption, [3.4.12](#)

RENT, [1.2.2](#)  
[3.4.44](#)

REPORT suboption, [3.4.24](#)

SCOL1F, [1.2.2](#)  
[3.4.48](#)

SCOL1V, [1.2.2](#)  
[3.4.50](#)

SCOL2F, [1.2.2](#)  
[3.4.49](#)

SCOL2V, [1.2.2](#)  
[3.4.51](#)

SEARCH, [3.4.45](#)

SEQUENCE, [3.4.48](#)  
[3.4.49](#)  
[3.4.50](#)  
[3.4.51](#)

SHOWINC, [1.2.2](#)  
[3.4.52](#)

SONF, [1.2.2](#)  
[3.4.46](#)

SONV, [1.2.2](#)  
[3.4.47](#)

SOURCE, [3.4.53](#)

SPILL, [1.2.2](#)  
[3.4.54](#)

SSCOMM, [1.2.2](#)  
[3.4.55](#)

START, [3.4.56](#)

SUM suboption, [3.4.63](#)

TARGET, [3.4.57](#)

TERM, [3.4.58](#)

TESTBLK, [1.2.2](#)  
[3.4.60](#)

TESTLIN, [1.2.2](#)  
[3.4.61](#)

TESTON, [1.2.2](#)  
[3.4.59](#)  
TESTPTH, [1.2.2](#)  
[3.4.62](#)  
TESTSYM, [1.2.2](#)  
[3.4.63](#)  
TRUNC suboption, [3.4.13](#)  
UPCONV, [1.2.2](#)  
[3.4.64](#)  
XREF, [3.4.65](#)  
components, [1.1.1](#)  
[1.1.1.1](#)  
CSECT option, [1.2.2](#)  
[3.4.14](#)  
CSP/AD  
  minimum release, [1.1.2.1](#)  
CSP/AE  
  minimum release, [1.1.2.1](#)  
customizing C/VSE, [3.0](#)

## D

---

DASD  
  storage requirements, [1.1.2.2](#)  
Debug Tool for VSE/ESA  
DECK option, [3.4.15](#)  
default library, [1.1.3](#)  
defaults  
  planning to customize, [1.2.2](#)  
defaults, compile-time options  
  descriptions, [3.4](#)  
DFSORT/VSE  
  minimum release, [1.1.2.1](#)  
distribution media, [1.1.1.1](#)  
distribution tape, [2.4.2](#)  
DL/I DOS/VS  
  minimum release, [1.1.2.1](#)  
DOS/VS Sort/Merge  
  minimum release, [1.1.2.1](#)

## E

---

EDCOPLNK link book, [3.1.2](#)  
EDCDP compiler phase  
  link-edit, [3.1](#)  
[3.1.2](#)  
EDCXIDEL sample job, [4.3](#)  
EDCXILDF sample job, [2.3.1](#)  
EDCXILNK sample job, [3.1.2](#)  
EDCXINDF sample job, [2.3.2](#)  
EDCXINST sample job, [2.4.2](#)  
EDCXIOP1 job step, [3.1.1](#)  
EDCXIOP2 job step, [3.1.1](#)  
EDCXIOPT sample job, [3.1](#)  
[3.1.1](#)  
EDCXIREM sample job, [4.3](#)  
EDCXIRET sample job, [4.2.3](#)  
EDCXISER sample job, [4.2.5.2](#)  
EDCXISMP sample source, [2.5](#)  
EDCXIVDF sample job, [2.3.1](#)  
EDCXIVP sample job, [2.5](#)

EDCXIVP1 job step, [2.5](#)  
EDCXIVP2 job step, [2.5](#)  
EDCXIVTC sample job, [2.3.2](#)  
EDCZ110A macro, [3.1](#)  
ENUM suboption, [3.4.4](#)  
EXECOPS option, [1.2.2](#)  
[3.4.16](#)  
EXPMAC option, [1.2.2](#)  
[3.4.17](#)  
EXTERN suboption, [3.4.5](#)

## F

---

feature numbers, [1.1.1.1](#)  
FLAG option, [1.2.2](#)  
[3.4.18](#)

## G

---

GENERAL suboption, [3.4.6](#)  
GONUMB option, [1.2.2](#)  
[3.4.19](#)  
GOTO suboption, [3.4.7](#)

## H

---

High Level Assembler  
  minimum release, [1.1.2.1](#)  
HWOPTSTR option, [1.2.2](#)  
[3.4.20](#)

## I

---

IBM Support Center, [1.1.5](#)  
IJSYSHF system history file, [2.4.2](#)  
INFILE option, [3.4.21](#)  
INIT suboption, [3.4.8](#)  
INLINE option, [3.4.22](#)  
INLNAUT option, [1.2.2](#)  
[3.4.23](#)  
INLNLIM option, [1.2.2](#)  
[3.4.26](#)  
INLNON option, [1.2.2](#)  
[3.4.22](#)  
INLNREP option, [1.2.2](#)  
[3.4.24](#)  
INLNTHR option, [1.2.2](#)  
[3.4.25](#)  
installation  
  allocate library space, [2.3](#)  
  allocate non-VSAM library space, [2.3.2](#)



- allocate VSAM library space, [2.3.1](#)
- backup original system, [2.2](#)
- condition codes and messages, [2.3.1](#)
  - [2.3.2](#)
  - [2.4.1.1](#)
  - [2.5](#)
- create library, [2.3.2](#)
- list directory entries, [2.4.2](#)
- messages, [2.4.1.1](#)
- retrace C/VSE, [2.4.2](#)
- summary of steps, [2.1.1](#)
- using a batch job, [2.4.2](#)
- using batch job, [2.4.2](#)
- using the interactive interface, [2.4.1](#)
- installation job stream
  - EDCXINST, [2.4.2](#)
- installation jobstream
  - EDCXIVP, [2.5](#)
- interactive interface
  - using to install C/VSE, [2.4.1](#)

## J

---

- job control
  - for allocating C/VSE library, [2.3.2](#)
  - for installing C/VSE, [2.4.2](#)

## L

---

- LANGLVL option, [1.2.2](#)
  - [3.4.27](#)
- language support
  - installing languages, [2.4.2](#)
  - selecting languages, [2.4.2](#)
- LE/VSE
  - C-specific base component, [1.1.2.1](#)
    - [2.5](#)
  - minimum release, [1.1.2.1](#)
- LIBR
  - LISTDIR, [1.1.3](#)
  - sublibrary, [2.5](#)
- LIBR librarian program
  - using to define C/VSE library space, [2.3.2](#)
- library
  - allocating space for C/VSE, [2.3](#)
- library requirements, [1.1.2.2](#)
- LINE suboption, [3.4.61](#)
- LINK
  - VSE system option, [3.4.38](#)
    - [3.4.40](#)
- LIST option, [3.4.28](#)
- LISTX
  - VSE system option, [3.4.28](#)
- LOCALE option, [1.2.2](#)
  - [3.4.30](#)
- LOCALEON option, [1.2.2](#)
  - [3.4.29](#)
- LONGNAME option, [1.2.2](#)
  - [3.4.31](#)
- LSEARCH option, [3.4.32](#)

# M

---

machine-readable material, [1.1.1.1](#)  
maintaining C/VSE, [4.0](#)  
MCOL1F option, [1.2.2](#)  
[3.4.33](#)  
MCOL1V option, [1.2.2](#)  
[3.4.35](#)  
MCOL2F option, [1.2.2](#)  
[3.4.34](#)  
MCOL2V option, [1.2.2](#)  
[3.4.36](#)  
MEMORY option, [1.2.2](#)  
[3.4.37](#)  
MSHP, [2.4.2](#)  
[4.2.3](#)  
INSTALL product, [2.4.2](#)  
retrace APARS, [4.2.3](#)  
RETRACE component, [2.4.2](#)  
retrace PTFs, [4.2.3](#)

# N

---

NAME option, [3.4.38](#)  
national language  
changing, [1.1.4](#)  
[1.2.4](#)  
[2.4.2](#)  
[3.3](#)  
installing, [2.4.2](#)  
NATLANG, [1.2.4](#)  
[3.3](#)  
NESTINC option, [1.2.2](#)  
[3.4.39](#)  
NLS  
See national language

# O

---

OBJECT option, [3.4.40](#)  
OFFSET option, [1.2.2](#)  
[3.4.41](#)  
OPT option, [1.2.2](#)  
[3.4.42](#)  
OPTION statements (JCL)  
CATAL, [3.4.40](#)  
DECK, [3.4.15](#)  
LINK, [3.4.40](#)  
LIST, [3.4.53](#)  
LISTX, [3.4.28](#)  
NAME, [3.4.38](#)  
TERM, [3.4.58](#)  
XREF, [3.4.65](#)  
optional licensed programs for C/VSE, [1.1.2.1](#)  
options  
IBM-supplied defaults, [3.4](#)

## P

---

PARM suboption, [3.4.9](#)  
PATH suboption, [3.4.62](#)  
phase name  
    automatic generation of, [3.4.38](#)  
PL/I for VSE/ESA  
    minimum release, [1.1.2.1](#)  
Planning for C/VSE, [1.0](#)  
PORT suboption, [3.4.10](#)  
PPCHECK suboption, [3.4.11](#)  
PPONLY option, [1.2.2](#)  
    [3.4.43](#)  
PPTRACE suboption, [3.4.12](#)  
preprocessor checking, [3.4.11](#)  
PTFs, [4.2.3](#)  
publications  
    C/VSE, [1.1.1.2](#)

## Q

---

QMF/VSE  
    minimum release, [1.1.2.1](#)

## R

---

re-installing C/VSE, [4.1](#)  
removing C/VSE, VSE, [4.3](#)  
RENT option, [1.2.2](#)  
    [3.4.44](#)  
REPORT suboption, [3.4.24](#)

## S

---

sample jobs  
    EDCXILDF, [2.3.1](#)  
    EDCXILNK, [3.1.2](#)  
    EDCXINDF, [2.3.2](#)  
    EDCXINST, [2.4.2](#)  
    EDCXIOPT, [3.1](#)  
        [3.1.1](#)  
    EDCXIVDF, [2.3.1](#)  
    EDCXIVP, [2.5](#)  
    EDCXIVTC, [2.3.2](#)  
sample source  
    EDCXISMP, [2.5](#)  
SCOL1F option, [1.2.2](#)  
    [3.4.48](#)  
SCOL1V option, [1.2.2](#)  
    [3.4.50](#)  
SCOL2F option, [1.2.2](#)

[3.4.49](#)  
SCOL2V option, [1.2.2](#)  
[3.4.51](#)  
SEARCH option, [3.4.45](#)  
SEQUENCE option, [3.4.48](#)  
[3.4.49](#)  
[3.4.50](#)  
[3.4.51](#)  
service updates, [1.1.5](#)  
Shared Virtual Area  
  loading compiler into, [1.2.3](#)  
    [3.2](#)  
  planning for, [1.2.3](#)  
  storage requirements, [1.2.3](#)  
SHOWINC option, [1.2.2](#)  
[3.4.52](#)  
SONF option, [1.2.2](#)  
[3.4.46](#)  
SONV option, [1.2.2](#)  
[3.4.47](#)  
SOURCE option, [3.4.53](#)  
SPILL option, [1.2.2](#)  
[3.4.54](#)  
SQL/DS  
  minimum release, [1.1.2.1](#)  
SSCOMM option, [1.2.2](#)  
[3.4.55](#)  
standard labels, [2.4.2](#)  
  IJSYSHF system history file, [2.4.2](#)  
START option, [3.4.56](#)  
storage  
  DASD requirements, [1.1.2.2](#)  
  SVA requirements, [1.2.3](#)  
SUBSET value, [1.1.5](#)  
SUM suboption, [3.4.63](#)  
supplied samples  
  EDCXIDEL, [4.3](#)  
  EDCXILNK, [3.1.2](#)  
  EDCXIOPT, [3.1.1](#)  
  EDCXIREM, [4.3](#)  
  EDCXIRET, [4.2.3](#)  
  EDCXISER, [4.2.5.2](#)  
  EDCXIVP, [2.5](#)  
SVA  
  See Shared Virtual Area  
system history file, [2.4.2](#)  
[4.2.3](#)

## T

---

tape format, [1.1.1.1](#)  
TARGET option, [3.4.57](#)  
TERM option, [3.4.58](#)  
TESTBLK option, [1.2.2](#)  
[3.4.60](#)  
TESTLIN option, [1.2.2](#)  
[3.4.61](#)  
TESTON option, [1.2.2](#)  
[3.4.59](#)  
TESTPTH option, [1.2.2](#)  
[3.4.62](#)  
TESTSYM option, [1.2.2](#)  
[3.4.63](#)  
tracing include files, [3.4.12](#)  
TRUNC suboption, [3.4.13](#)

## U

---

UPCONV option, [1.2.2](#)  
[3.4.64](#)  
UPGRADE value, [1.1.5](#)

## V

---

verification program, [2.5](#)  
verify applied service, [4.2.6](#)  
verify installation  
    EDCXIVP, [2.5](#)  
VSE/ESA  
    minimum release, [1.1.2.1](#)  
    system requirements, [1.1.2](#)

## W

---

WXTRN, [3.1.2](#)

## X

---

XREF option, [3.4.65](#)

---

# BACK\_1 Communicating Your Comments to IBM

IBM C for VSE/ESA  
Installation and Customization Guide  
Release 1

Publication No. GC09-2422-00

If there is something you like--or dislike--about this book, please let us know. You can use one of the methods listed below to send your comments to IBM. If you want a reply, include your name, address, and telephone number. If you are communicating electronically, include the book title, publication number, page number, or topic you are commenting on.

The comments you send should only pertain to the information in this book and its presentation. To request additional publications or to ask

questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

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

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give it to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments by FAX, use this number:
  - United States and Canada: 416-448-6161
  - Other countries: (+1)-416-448-6161
- If you prefer to send comments electronically, use the network ID listed below. Be sure to include your entire network address if you wish a reply.
  - Internet: torrcf@vnet.ibm.com
  - IBMLink: toribm(torrcf)
  - IBM/PROFS: torolab4(torrcf)
  - IBMMAIL: ibmmail(caibmwt9)

---

## COMMENTS Readers' Comments -- We'd Like to Hear from You

IBM C for VSE/ESA  
Installation and Customization Guide  
Release 1

Publication No. GC09-2422-00

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

Legend:

- 1 Very satisfied
- 2 Satisfied
- 3 Neutral
- 4 Dissatisfied
- 5 Very dissatisfied

	1	2	3	4	5
Overall satisfaction					

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

	1	2	3	4	5
Accurate					
Complete					
Easy to find					
Easy to understand					
Well organized					
Applicable to your tasks					

Please tell us how we can improve this book:

IBM Canada Ltd. Laboratory  
Information Development  
2G/345/1150/TOR  
1150 EGLINTON AVENUE EAST  
NORTH YORK ONTARIO CANADA M3C 1H7

Name . . . . . \_\_\_\_\_  
Company or Organization \_\_\_\_\_  
Address . . . . . \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
Phone No. . . . . \_\_\_\_\_

**IBM Library Server Print Preview**

DOCNUM = GC09-2422-00  
DATE TIME = 10/23/96 11:01:02  
BLDVERS = 1.2  
TITLE = C/VSE V1R1 Installation and Customization Guide  
AUTHOR =  
COPYR = © Copyright IBM Corp. 1996  
PATH = /home/webapps/epubs/htdocs/book

---