Licensed Program Specifications

IBM COBOL for VSE/ESA
Program Number 5686-068

IBM* COBOL for VSE/ESA* (COBOL/VSE) is based on IBM SAA* AD/Cycle* COBOL/370* Release 1 and incorporates the VSE support provided in VS COBOL II Version 1 Release 4.

COBOL/VSE and its generated object programs, execute under the following specific operating systems and subsystems, and their follow-on releases: VSE/ESA Version 1 Release 4, VSE/ESA Version 2 Release 1, and CICS/VSE* Version 2 Release 3.

COBOL/VSE uses Language Environment* for VSE/ESA (LE/VSE), a prerequisite product, as its run-time environment. LE/VSE must be ordered separately.

The COBOL/VSE compiler is a single licensed product package, with support for the functions described below.

Support for LE/VSE

The COBOL/VSE compiler generates object code to be run with LE/VSE, which provides a common run-time environment for all conforming high-level languages, including COBOL/VSE.

LE/VSE provides the following functions:

- A common set of protocols for interlanguage communications
- A common set of services that COBOL programs can call, including:
  - Date and time services
  - Dynamic storage services
  - Condition handling services
  - Message handling services
  - National language support services
  - General services, including a formatted dump capability
- Space tuning at run time

COBOL/VSE can also generate programs that, when run in the LE/VSE environment, can be reentrant and take advantage of extended, 31-bit addressing.

Reentrant Programs: COBOL/VSE programs can be compiled to be reentrant. A reentrant program (or a reentrant subroutine) can be placed in the shared virtual storage where one copy of the program can satisfy multiple, concurrent requests.

Extended Addressing: Extended addressing allows programs and data areas to be larger than they can be with DOS/VS COBOL. COBOL/VSE supports the extended addressing capabilities by:

- Allowing most of the compiler to run in 31-bit addressing mode and to be loaded above 16 megabytes.
- Allowing programs compiled by COBOL/VSE to be executed in the 31-bit addressing mode of extended architecture processors. Certain options or environments may preclude this capability.
- Allowing COBOL programs that run in 31-bit mode to be used with CICS*.
- Allowing DOS/VS DL/I calls from COBOL/VSE programs executing in addresses above 16 megabytes.

Industry Standards

COBOL/VSE supports the following industry standards in the VSE/ESA environment, as understood and interpreted by IBM as of September 1989:

1. ISO 1989:1985, Programming Languages - COBOL

   ISO 1989/Amendment 1, Programming Languages - COBOL - Amendment 1: Intrinsic function module.

   ISO 1989:1985 is identical to X3.23-1985, American National Standard for Information Systems - Programming Language - COBOL. ISO 1989/Amendment 1 is identical to X3.23a-1989, American National Standard for Information Systems - Programming Language - Intrinsic Function Module for COBOL.

   For supported modules, see item 2 below.

2. X3.23-1985, American National Standard for Information Systems - Programming Language - COBOL.

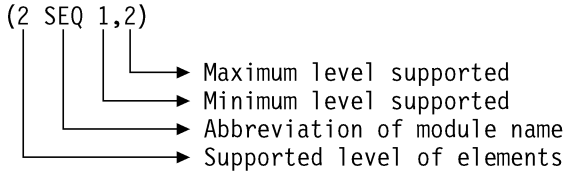
   X3.23a-1989, American National Standard for Information Systems - Programming Language - Intrinsic Function Module for COBOL.

   COBOL/VSE supports all required modules at the intermediate level. It also supports all required modules at the high level with the exception of the following language features:

   - EXTEND phrase of the OPEN statement
   - REVERSED phrase of the OPEN statement
   - OF/IN phrase of the COPY statement

   See Level 2 Restrictions below.

   In the following list, the shorthand notation describing module levels is shown in parentheses. For example, to summarize module information for sequential input/output, the shorthand notation is (2 SEQ 1,2).



   - Nucleus (2 NUC 1,2)
   - Sequential I-O (1 SEQ 1,2) file.

     Level 2 restriction: the EXTEND phrase of the OPEN statement is not supported except for VSAM sequential files.

     Level 2 restriction: the REVERSED phrase of the OPEN statement does not cause file positioning, and is only applicable to tape files.

   - Relative I-O (2 REL 0,2)
   - Indexed I-O (2 INX 0,2)
   - Sort-Merge (1 SRT 0,1)
   - Inter-Program Communication (2 IPC 1,2)
   - Source Text Manipulation (1 STM 0,2)

     Level 2 restriction: the OF/IN phrase of the COPY statement is treated as documentation.

   In addition, the following levels of optional modules are supported:

   - Intrinsic Functions (1 ITR 0,1)
   - Debug (1 DEB 0,2)
   - Segmentation (2 SEG 0,2)

   The following optional modules of the standard are not supported:

   - Report Writer
   - Communications
   - Debug (2 DEB 0,2)

3. FIPS Publication 21-3, Federal Information Processing Standard 21-3 , COBOL high subset.

4. International Reference Version of the ISO 7-bit code defined in *International Standard 646, 7-Bit Coded Character Set for Information Processing Interchange*.

5. The 7-bit coded character sets defined in *American National Standard X3.4-1977, Code for Information Interchange*.

Under CICS, the following language elements of X3.23-1985 are not supported:

ACCEPT
CLOSE
DELETE
DISPLAY
MERGE
OPEN
READ
RERUN
REWRITE
SORT which requires COBOL to perform I/O

START
STOP 'literal'
WRITE
USE declaratives (except USE FOR DEBUGGING)
ENVIRONMENT DIVISION and FILE SECTION when entries relate to data management.

NOTES:

The term "COBOL 85 Standard" is used in this book to refer to the combination of the following standards:

- ISO 1989:1985, Programming Languages - COBOL

  ISO 1989/Amendment 1, Programming Languages - COBOL - Amendment 1: Intrinsic function module.

- X3.23-1985, American National Standard for Information Systems - Programming Language - COBOL.

  X3.23a-1989, American National Standard for Information Systems - Programming Language - Intrinsic Function Module for COBOL.

The term "COBOL 74 Standard" is used in this book to refer to the following standards:

- X3.23-1974, American National Standard for Information Systems - Programming Language - COBOL.
- ISO 1989:1978, Programming Languages - COBOL

Note: The ISO Standards are equivalent to the American National Standards.

The following options are **required** to support the above standards:

Compiler Options LE/VSE Run-Time Options

ADV	AIXBLD
DYNAM	TRAP(ON)
FLAGSTD(H)	
LIB	
NOCMPR2	
NOCURRENCY	
NODBCS	
NOFASTSRT	
NOFLAGMIG	
NOFLAGSAA	
NONUMBER	
NOSEQUENCE	
NUMPROC(NOPFD) or NUMPROC(MIG)	
QUOTE	
TRUNC(STD)	
NOWORD	
ZWB	

The following LE/VSE run-time options are used in support of the standards: UPSI, DEBUG, and NODEBUG.

COBOL/VSE Language Features
The COBOL/VSE language, in support of the industry standards listed above, together with IBM language extensions, includes support for:

- Intrinsic functions
- Structured programming
- Language Extensions to LE/VSE
- Double-byte character set (DBCS)
- Feedback from VSAM return codes
- File and data sharing
- CICS language Interface
- Associated Data (ADATA)

Intrinsic Function Support: The intrinsic function support provides mathematical, statistical, financial, string handling, date, and time functions. Intrinsic functions provide the capability to refer to a data item whose value is derived automatically at the time of reference. The ALL subscript can be used with intrinsic functions to perform calculations on an entire table with one COBOL statement.

Structured Programming Support: This support includes these constructs to aid in structured programming:

- Nested programs and nested copy statements
- Inline PERFORM statements
- Explicit scope terminators
- The EVALUATE statement

Language Extensions to Support LE/VSE: In support of the LE/VSE callable services, COBOL/VSE adds support for a new procedure-pointer data type.

Double-Byte Character Set (DBCS):
COBOL/VSE accepts characters in the double-byte character set (DBCS). DBCS uses the same identifiers and literals as the former Extended Graphic Character Set (EGCS). In addition, COBOL/VSE allows reference modification of DBCS data items and allows DBCS literals in the TITLE statement. DBCS characters are also allowed in nonnumeric literals.

Feedback from VSAM Return Codes: This support gives users access to VSAM return codes to obtain detailed information about their VSAM input/output requests.

File and Data Sharing: COBOL/VSE allows a program to share files and data with other programs in a run unit by using the EXTERNAL attribute. Nested programs allow the sharing of files and data by use of the GLOBAL attribute. Data can also be protected in a nested program by **not** specifying the GLOBAL attribute.

CICS Language Interface: COBOL/VSE subprograms can contain CICS commands. In addition, pointer variables and the ADDRESS OF and LENGTH OF special registers, in conjunction with the CALL statement, can be used to simplify CICS COBOL application development.

Associated Data File: This feature allows users to produce, via the ADATA compiler option, a file containing program data. From this file (SYSADAT) information about the compiled program can be extracted. The SYSADAT file provides a clearly differentiated programming interface (CDPI) to the compiler output. Utilities that previously parsed the compiler output list can now make use of SYSADAT records.

Compiler Features

The COBOL/VSE compiler includes function to:

- Optimize generated code
- Produce diagnostics and listings
- Aid debugging
- Support batch compiling
- Enable COBOL user exits
- Support fast sorting
- Support national language
- Control installation

Object Code Optimization: Run time of the object programs may be reduced; the amount of storage used by the programs may also be reduced.

Diagnostics and Listings: COBOL/VSE provides features that can assist in the tasks of developing and diagnosing programs:

- Embedded diagnostic messages (messages appear in the listing following the statement that caused them)
- Embedded data map report
- Embedded cross-reference information
- Consolidation of messages that were caused by the same error occurring many times
- Alphabetized cross-reference listing
- Nested program map
- Data Division map
- Source statement listing

Aiding Debugging: Compiler options can be specified to generate object code that is set up to:

- Get a symbolic dump of the Data Division of COBOL programs
- Verify that subscripted data items and variable-length data items are addressed only within the data space allocated.

Batch Compiling: A sequence of separate COBOL programs can be compiled with a single invocation of the compiler, and the resultant object programs can then be link-edited either into one phase, or into multiple phases.

User Exits: The EXIT compiler option provides user exits for user-supplied custom I/O routines.

Fast Sorting: COBOL/VSE provides the FASTSRT compiler option, which provides the capability for DOS/VS Sort/Merge II or DFSORT/VSE to perform the I/O on COBOL sort input and sort output files. For certain COBOL/VSE programs run with LE/VSE, this can provide faster run-time sorting.

National Language Support: COBOL/VSE allows compiler diagnostics, run-time messages, and listing headings to be displayed in a language other than English. A separate language feature is required to implement the translated messages.

Installation Control: An installation can enforce "in-house" programming standards through reserved word control and compiler option control. For example, use of the *GO TO* or *ALTER* verb can be prohibited.

Compatibility Features

Operating System Compatibility: This feature allows object modules produced with COBOL/VSE to be migrated to VM/ESA* and MVS/ESA* where they can execute compatibly with each other, if LE/370 is used.

Compatibility with DOS/VS COBOL: COBOL/VSE produces equivalent run-time results with DOS/VS COBOL Release 3.0, or later, for all supported language elements of X3.23-1974, American National Standard Programming Language COBOL, when using the CMPR2 compiler option. Additionally, many IBM extensions supported by DOS/VS COBOL Release 3.0, or later, are supported in COBOL/VSE. Any invalid language that may have executed in DOS/VS COBOL is not guaranteed to work in COBOL/VSE.

Object modules produced from valid programs compiled with DOS/VS COBOL Release 3.0 (under VSE), or later, will execute compatibly with the LE/VSE Library.

Compatibility is not maintained for programs dependent on a specific implementation of the DOS/VS COBOL compiler, such as specific code sequences, program structure, and internal control block formats.

In COBOL/VSE, many 1968 standard language elements and obsolete extensions are removed. Obsolete semantics will be diagnosed as errors in COBOL/VSE.

Compatibility with VS COBOL II Release 3 or Later: COBOL/VSE provides upward source program compatibility with VS COBOL II Release 3 or later, with three exceptions resulting from conformance with the published ANSI interpretation of X3.23-1985 COBOL standard. These exceptions are:

- Different treatment of blank lines and comment lines appearing in pseudo-text-1 of COPY REPLACING and REPLACE
- Precedence of USE procedures in contained programs
- Maximum length receiver rule applying to reference-modified groups that are variable length and contain their own ODO (OCCURS DEPENDING ON) clause.

Programs using any of these language elements and recompiled on COBOL/VSE can produce different run-time results than with Release 3.2 and 4 of VS COBOL II.

Also, because of changes to some of the compiler listing formats in Release 4, if the Release 3.2 program depended on these formats, results can vary.

Compatibility with VS COBOL II Release 2: Implementation of support for the COBOL 85 STANDARD, beginning with VS COBOL II Release 3, causes some language elements to behave in a different manner than in VS COBOL II Release 2. Any valid Release 2 program that is compiled using COBOL/VSE with the CMPR2 compiler option specified will provide, when run with LE/VSE, equivalent run-time results. However, because of changes in the compiler listing formats, if the Release 2 program depends on the listing format, results can vary.

Link-editing Requirements: If any part of a phase is recompiled with COBOL/VSE, the phase must be relinked with the LE/VSE library. Note that with COBOL/VSE, relinked phases are **not** self-contained.

CICS Considerations: For migration from DOS/VS COBOL to COBOL/VSE, CICS Command Level programs must be compiled with the RENT and NODYNAM options provided by the CICS Translator. All references to COBOL BLL cells must be replaced or removed.

Specified Operating Environment

Programming Requirements

COBOL/VSE application programs can be run in all environments supported by LE/VSE. In conjunction with LE/VSE Release 1, object programs generated by COBOL/VSE can be run under, or with, the following IBM licensed programs and their subsequent releases unless otherwise announced by IBM. Later releases of LE/VSE may require or only support later releases and versions of these products; refer to the Program-

ming Requirements section of the LE/VSE Licensed Program Specifications.

- VSE/ESA Version 1 Release 4 (5750-ACD)
- VSE/ESA Version 2 Release 1 (5690-VSE)
- CICS/VSE Version 2 Release 3 (5686-026)
- SQL/DS* Version 3 Release 4
- DOS/VS Sort/Merge II Version 2 Release 5 (5746-SM2)
- DFSORT/VSE Version 3 Release 1 (5746-SM3)
- DL/I DOS/VS Version 1 Release 10 (5746-XX1)
- High Level Assembler/MVS & VM & VSE Release 1 (5696-234)
- High Level Assembler/VSE Release 2 (5697-013)

For installation under VSE/ESA, the IBM Maintain System History Program (MSHP) is required.

For installation of maintenance modifications in certain modules and customization of certain default macros under VSE/ESA, High Level Assembler Release 1 is required.

Machine Requirements

The COBOL/VSE compiler and its generated object programs will run on IBM system configurations that support the programming systems specified above.

Most of the compiler can be executed in 31-bit mode and can be loaded into virtual addresses above 16 megabytes.

IBM devices supported by the VSAM and SAM access methods on the above systems can be used with object programs produced by the COBOL/VSE compiler.

The COBOL/VSE compiler requires about 760K bytes of virtual storage. These 760K bytes are for the compiler and do not include space for operating system overhead or other programs.

Licensed Program Materials Availability

Restricted materials—No. This licensed program is available without source licensed program materials. It is available in object code.

Supplemental Terms

Designated Machine Identifications

Designated Machine Identifications Required-Yes.

Testing Period

- Basic License: One month
- DSLO License: Not applicable

Installation/Location License

Not applicable. A separate license is required for each machine on which the licensed program will be used.

Usage License

Not applicable

Type/Duration of Program Services

- Central Service
- Until discontinued by IBM with a minimum of twelve months written notice

Softcopy Publications

The program that IBM licenses may include licensed publications in displayable or source form. Except as provided in this section, the terms and conditions of the license agreement with IBM apply to these publications and to any copies that you make from them.

The licensed publications may be used in displayable or source form on all machines designated for this program. The licensed publications may also be copied and used on other machines in support of authorized use of this program.

To support authorized use of the Program, printed copies of the displayable or source material may be made if the copyright notice and any other legend of ownership is reproduced on each copy or partial copy.

Warranty

This program is warranted as specified in the IBM license.

Licensed Program Specifications may be updated from time to time and such updates may constitute a change in specifications.

For Distributed Systems License Option (DSLO) Licenses, warranty service, if any, will be provided only through the Basic License location.

Following the discontinuance of all program services, this program will be provided “As Is” as specified in the IBM license.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service 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.

Any other documentation with respect to this licensed program, including any documentation referenced herein, is provided for reference purposes only and does not extend or modify these specifications.

GC26-8069-00

April 1995

Printed in U.S.A.