SC28-6481-2
File No. S370-24

Program Product

IBM OS/VS COBOL Compiler and Library Installation Reference Material

Program Numbers 5740-CB1 (Compiler and Library)
5740-LM1 (Library Only)

Release 2

Third Edition (May 1983)

This is a major revision of, and makes obsolete, SC28-6481-1, and its technical newsletter, SN20-9270.

This edition applies to Release 2 of OS/VS COBOL Compiler and Library, Program Products 5740-CB1 (Compiler and Library) and 5740-LM1 (Library only), and to any subsequent releases until otherwise indicated in new editions or technical newsletters.

The changes for this edition are summarized under "Summary of Amendments" following the preface. Specific changes are indicated by a vertical bar to the left of the change. These bars will be deleted at any subsequent republication of the page affected. Editorial changes that have no technical significance are not noted.

Changes are periodically made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/370 and 4300 Processors Bibliography, GC20-0001, for the editions that are applicable and current.

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 program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

Publications are not stocked at the address given below; requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, P.O. Box 50020, Programming Publishing, San Jose, California, U.S.A. 95150. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

PREFACE

This manual is designed for system programmers and planners who supervise the generation and maintenance of an organization's operating system. It contains material for installing the OS/VS COBOL Compiler and Library and is to be used in conjunction with the OS/VS COBOL Program Directory that applies to your system.

MANUAL ORGANIZATION

The manual has four major sections:

- **Introduction** contains high-level information for installing OS/VS COBOL. Included is a brief description of the OS/VS COBOL Compiler and Library.
- **System Requirements** contains the storage estimates and machine requirements needed for the compiler and library to be installed on OS/VS1, OS/VS2 (with or without TSO), or CMS.
- **Specifying the Compiler Default Options** describes the VSCBL1 macro instruction that you use during installation.
- **OS/VS COBOL Installation** describes the installation process as it applies to each of the operating systems.

There are also two appendixes:

- Appendix A lists books that contain device information.
- Appendix B lists the Library modules.

INDUSTRY STANDARDS

The OS/VS COBOL Compiler and Library, Release 2.4, is designed according to the specifications of the following industry standards, as understood and interpreted by IBM as of April, 1976:

- The highest level of American National Standard COBOL, X3.23-1974 (excepting the Report Writer module). American National Standard COBOL, X3.23-1974, is compatible with and identical to International Organization for Standardization/Draft International Standard (ISO/DIS) 1989-COBOL.
- The highest level of American National Standard COBOL, X3.23-1968. American National Standard COBOL, X3.23-1968, is compatible with and identical to ISO/R 1989-1972 Programming Language COBOL.

A number of IBM extensions are also implemented.

RELATED PUBLICATIONS

OS/VS COBOL PUBLICATIONS

- IBM OS/VS COBOL Compiler and Library Programmer's Guide, SC28-6483
- IBM VS COBOL for OS/VS, GC26-3857

COBOL INTERACTIVE DEBUG PUBLICATION

- IBM OS COBOL Interactive Debug Terminal User's Guide and Reference, SC28-6465

SMP4 PUBLICATION

- OS/VS System Modification Program (SMP) System Programmer's Guide, GC28-0673

OS/VS1 PUBLICATIONS

- OS/VS1 Access Method Services, GC26-3840
- OS/VS1 Data Management Services Guide, GC26-3874
- OS/VS1 JCL Reference, GC24-5099
- OS/VS1 JCL Services, GC24-5100
- OS/VS1 Utilities, GC26-3901
- OS/VS Sort/Merge Installation Reference Material, SC33-4034
- OS/VS Virtual Storage Access Method (VSAM) Programmer's Guide, GC26-3838

OS/VS2 PUBLICATIONS

- OS/VS2 Access Method Services, GC26-3841
- OS/VS2 MVS Data Management Services Guide, GC26-3875
- OS/VS2 MVS JCL, GC28-0692
- OS/VS2 MVS Utilities, GC26-3902
- OS/VS Sort/Merge Installation Reference Material, SC33-4034
- OS/VS Virtual Storage Access Method (VSAM) Programmer's Guide, GC26-3838

VM/370 AND CMS PUBLICATIONS

- IBM Virtual Machine Facility/370: CP Command Reference for General Users, GC20-1820
- IBM Virtual Machine Facility/370: CMS Command and Macro Reference, GC20-1818
- IBM Virtual Machine Facility/370: CMS User's Guide, GC20-1819
- IBM Virtual Machine Facility/370: Planning and System Generation Guide, GC20-1801
- IBM Virtual Machine Facility/370: System Programmer's Guide, GC20-1807
- IBM Virtual Machine Facility/370: Operator's Guide, GC20-1806

VM/SP AND CMS PUBLICATIONS

- IBM Virtual Machine/System Product: CP Command Reference for General Users, SC19-6211

- IBM Virtual Machine/System Product: CMS Command and Macro Reference, SC19-6209

- IBM Virtual Machine/System Product: CMS User's Guide, SC19-6210

- IBM Virtual Machine/System Product: Planning and System Generation Guide, SC19-6201

- IBM Virtual Machine/System Product: System Programmer's Guide, SC19-6203

SUMMARY OF AMENDMENTS

MAY 1983

New Installation Procedure

IBM OS/VS COBOL is installed using the OS/VS System Modification Program (SMP). The installation procedures have been rewritten to reflect the change.

CONTENTS

PART 1. PLANNING FOR INSTALLATION

This part contains the information you will need to plan for the installation of OS/VS COBOL.

- **Introduction** on page 2

  Describes the COBOL product.

- **System Requirements** on page 3

  Defines the requirements for OS/VS COBOL.

- **Specifying the Compiler Default Options** on page 6

  Provides the format and description of the VSCBL1 macro instruction.

INTRODUCTION

The OS/VS COBOL Compiler and Library is a program product that runs under OS/VS1, OS/VS2 (with or without TSO), MVS/XA (24-bit mode only), and CMS. The Compiler and Library are available as a single program product; the Library is also available separately. Each program product is distributed on its own tape containing the necessary modules, as well as test programs for verifying the installation procedures.

OS/VS COBOL COMPILER

The OS/VS COBOL Compiler processes programs written in the language described in IBM VS COBOL for OS/VS, and produces object modules that are suitable as input to the linkage editor for subsequent execution on IBM System/370, IBM 303x Processors, IBM 308x Processors, and IBM 43xx Processors. The Compiler contains the following features:

- Optimized object code
- Symbolic debugging
- The COBOL library management facility
- Dynamic subprogram linkage
- Support for teleprocessing
- Syntax-checking compilation
- Support for VSAM
- Enhanced compiler listing facilities
- Verb usage statistics
- Support for the IBM 3886 Optical Character Reader
- Enhanced IBM Sort/Merge support
- A Federal Information Processing Standard (FIPS) flagger

For information on using the compiler after it is installed, see IBM OS/VS COBOL Compiler and Library Programmer's Guide.

OS/VS COBOL LIBRARY

The OS/VS COBOL Library contains routines to perform frequently-required object time operations, including input/output routines, conversion routines, arithmetic verb routines, sort interface routines, debugging routines, and other verb routines (such as SEARCH).

SYSTEM REQUIREMENTS

This section describes the storage estimates and machine configuration required to support the OS/VS COBOL Compiler under OS/VS1, OS/VS2 (with or without TSO), MVS/XA (24-bit mode only), and CMS.

STORAGE ESTIMATES

The OS/VS COBOL compiler requires 128K (131072) bytes of virtual storage. During compilation, up to six work files may be necessary; therefore, auxiliary storage is required. During link editing, modules produced by the compiler may require subroutines from the OS/VS COBOL Library. In addition, subroutines from this library may be required during execution; therefore, the library must be part of the system used to control execution.

AUXILIARY STORAGE REQUIREMENTS

The storage requirements needed for the compiler and library are contained in the OS/VS COBOL Program Directory. For a list of manuals containing storage requirements, see Appendix A, "Publications."

EXECUTION-TIME STORAGE REQUIREMENTS

The amount of main storage at execution time must be sufficient to accommodate at least:

- The control program
- Data management support
- The load modules to be executed

The input/output device requirements for execution of the problem program are determined from specifications made in the source program.

MACHINE REQUIREMENTS

The machine requirements for OS/VS COBOL are as follows:

Item	Requirements
Installation-Time Machine Requirements	IBM System/370 Processors, IBM 303x Processors, IBM 308x Processors, or IBM 43xx Processors that can support the OS/VS1, OS/VS2 (with or without TSO), MVS/XA (24-bit mode only), or CMS environments. (See also "System Requirements" on page 3.) Space available on a private library for the OS/VS COBOL program product.
Compile-time Machine Requirements	Any processing unit supported by CMS, OS/VS1, OS/VS2, or MVS/XA. plus four utility data sets: SYSUT1, SYSUT2, SYSUT3, and SYSUT4. An additional data set, SYSUT5, is required when either the execution statistics feature, Symbolic Debugging, or the COBOL Interactive Debug Program Product, is used. This data set must be available during compile time and at execution time if debugging output is desired. An additional data set, SYSUT6, is required if the Federal Information Processing Standard (FIPS) flagger is used. These data sets require at least one direct-access storage device for residence of the operating system and SYSUT1. Both the operating system and SYSUT1 can reside on the same volume. SYSUT2, SYSUT3, SYSUT4, SYSUT5, and SYSUT6 can reside on tape or direct-access storage. If they reside on tape, there must be a tape volume for each data set. I/O devices used by the compiler.
Execution-time Machine Requirements	Any processing unit supported by CMS, OS/VS1, OS/VS2, or MVS/XA. I/O devices used by the object program during execution.
Devices Supported	Under OS/VS1, OS/VS2, MVS/XA, and CMS, IBM devices supported by the BSAM, BDAM, ISAM (except under CMS), QSAM, and VSAM access methods can be used by object programs produced by the OS/VS COBOL Compiler when used in conjunction with the OS/VS COBOL Library. OS/VS COBOL supports the IBM 3375 and 3380 Direct Access Storage Devices (DASD). OS/VS COBOL supports the IBM 3886 Optical Character Reader, and the IBM 3505 and 3525 multifunction Card Devices. The use of the 3886 requires at least OS/VS1 Release 3 or OS/VS2 Release 2.

SOFTWARE REQUIREMENTS

SMP4 REQUIREMENTS

The System Modification Program 4 (SMP4) is required to install OS/VS COBOL.

TSO REQUIREMENTS

Operation of the compiler under TSO requires the minimum TSO configuration.

VSAM REQUIREMENTS

The use of the Release 2 COBOL VSAM features requires at least VSAM Release 2.

COMMUNICATIONS (TCAM) REQUIREMENTS

Using the Communications Feature of COBOL requires the minimum operating system configuration for the fifth release of TCAM. The TCAM library, SYS1.TELCMLIB, is required in the link-edit step when running with NORESIDENT in effect and testing for TCAM or BSAM compatibility. This same TCAM library is required during object time when the user is running programs with RESIDENT in effect.

SORT/MERGE FEATURE REQUIREMENTS

Full use of the Sort/Merge feature of COBOL requires the presence of the IBM OS/VS Sort/Merge program product, Program Number 5740-SM1, at execution time. The system requirements of that program are given in OS/VS Sort/Merge Installation Reference Material.

SPECIFYING THE COMPILER DEFAULT OPTIONS

Before installing OS/VS COBOL, you need to define the compiler options for your installation. You may choose either to use the default options, as supplied by IBM, or to tailor the options to your needs. Compiler options are specified and altered by using the VSCBL1 macro instruction. Specify the macro instruction and store under member name IKFCBL01 of the SYS1.PPOPTION data set.

- To use the IBM-supplied default options, merely enter:

Syntax

```
VSCBL1
END
```

- To modify the options, enter the VSCBL1 macro with the operands you wish to change.

  For specifics concerning this step, see the Release 2.4 OS/VS COBOL Program Directory, section "Specifying Defaults for IBM OS/VS COBOL Compiler and Library Options."

The syntax (or format) of the VSCBL1 macro instruction is shown on the next page. The IBM-supplied default values are underlined. To change the default values, specify the VSCBL1 macro instruction with the operands that define the options to be changed.

The VSCBL1 macro instruction can be coded in columns 2 through 71 of a statement. If a statement exceeds column 71, it can be continued onto one or more additional statements; a nonblank character in column 72 indicates a continuation. The macro instruction can be interrupted after any comma that separates the operands. Coding on the continuation statement must begin in column 16.

The following conventions are used to illustrate the syntax of the macro instruction:

- UPPERCASE letters and numbers must be entered exactly as shown.
- Lowercase letters and words represent variables for which you must substitute information or values.
- Items or groups of items within brackets, [ ], are optional and may be omitted. The lack of brackets indicates that an item must be coded.
- The vertical stroke (often called the OR sign), |, indicates that a choice is to be made. Code only one of the items separated by the vertical stroke.
- Braces, { }, delimit the items that the vertical stroke applies to.
- An item that is underlined is the IBM-supplied default; the system will automatically assume that the default is the programmer's choice when none of the items are coded.

When the VSCBL1 macro instruction is invoked during installation of the compiler, a message indicates the options specified or the defaults accepted. If you have specified an invalid value, a message with a severity level of 5 identifies it, and the IBM-supplied default is indicated. A message with a severity level of 5 also appears when any conflicting values (such as XREF and SXREF) have been specified. When either of these messages appears, the installation of the compiler cannot proceed. You must specify the VSCBL1 macro instruction with the correct values and rerun the installation job.

Note: The VSCBL1 macro instruction statement(s) must be followed by an END statement.

Syntax

```
VSCBL1    [BATCH={NOBATCH|BATCH}]
          [,BUF=number]
          [,CALMOD={NODYNAM|DYNAM}]
          [,CMPMODE={NOSYNTAX|SYNTAX|CSYNTAX}]
          [,CNDENSE={NOCLIST|CLIST}]
          [,COUNTL={NOCOUNT|COUNT}]
          [,DATAMAP={NODMAP|DMAP}]
          [,DUMP={NODUMP|DUMP}]
          [,ENDJOB={NOENDJOB|ENDJOB}]
          [,FLOW={NOFLOW|FLOW}]
          [,FLOWSIZ={99|nn}]
          [,LIBRARY={NORESIDENT|RESIDENT}]
          [,LINECNT={60|lines}]
          [,LITCHAR={QUOTE|APOST}]
          [,LSTCOMP={NOLST|LSTCOMP}]
          [,LSTDECK={NOFDECK|FDECK}]
          [,LSTONLY={NOLST|LSTONLY}]
          [,LSTPCH={NOCDECK|CDECK}]
          [,LSTPRC2={LCOL2|LCOL1}]
          [,LST132={L132|L120}]
          [,LVL={NOLVL|A|B|C| D}]
          [,MSGLEV={FLAGW|FLAGE}]
          [,NAME={NONAME|NAME}]
          [,NEWADV={ADV|NOADV}]
          [,NUM={NONUM|NUM}]
          [,OPT={NOOPT|OPT}]
          [,PROCMAP={NOPMAP|PMAP}]
          [,PUNCH={NODECK|DECK}]
          [,REFLIST={NOXREF|XREF}]
          [,SEQCHK={SEQ|NOSEQ}]
          [,SIZE={131072|size}]
          [,SORLIST={SOURCE|NOSOURCE}]
          [,SPACE={SPACE1|SPACE2|SPACE3}]
          [,SREFLST={NOSXREF|SXREF}]
          [,STATE={NOSTATE|STATE}]
          [,STD={LANGLVL(2)|LANGLVL(1)}]
          [,SUPPRES={NOSUPMAP|SUPMAP}]
          [,SYMDMP={NOSYMDMP|SYMDMP}]
          [,SYSLIB={NOLIB|LIB}]
          [,SYSOUT={SYST|SYSx}]
          [,TERM={NOTERM|TERM}]
          [,TEST={NOTEST|TEST}]
          [,TRNCATE={NOTRUNC|TRUNC}]
          [,TYPERUN={LOAD|NOLOAD}]
          [,VERB={VERB|NOVERB}]
          [,VRBREF={NOVBREF|VBREF}]
          [,VRBSUM={NOVBSUM|VBSUM}]
          [,ZWHENB={ZWB|NOZWB}]
END
```

BATCH=
specifies the compilation of multiple programs with a single invocation of the compiler. BATCH should not be specified when SYMDMP is in effect.

BATCH
specifies multiple compilation.

NOBATCH
specifies no multiple compilation.

BUF=

specifies the number of bytes of main storage to be used for buffer allocation by the COBOL compiler.

number

is an integer from 4096 to 999999. This value must be included in the value given for the SIZE parameter. If BUF is omitted and SIZE is specified, the value of BUF is calculated as:

$$\frac{\text{SIZE}-98304}{4} + 4096$$

Note: If both BUF and SIZE are omitted, a value of 12288 is assumed for BUF.

CALMOD=

DYNAM

specifies that the CALL literal statement is to be treated as a dynamic CALL and all user subprograms are to be dynamically loaded at object time. When DYNAM is specified, RESIDENT is also always in effect.

NODYNAM

specifies that all user subprograms invoked with a CALL literal statement are to be included in the load module produced at link-edit time.

CMPMODE=

SYNTAX

specifies that a syntax-checking compilation is to be performed.

CSYNTAX

specifies that a conditional syntax-checking compilation is to be performed. A full compilation is produced when no message exceeds the warning (W) or conditional (C) level.

NOSYNTAX

specifies that a full compilation is to be performed.

When SYNTAX is in effect, the following options are automatically negated at compile time:

XREF
SXREF
PMAP
DMAP
LOAD
DECK
NOSUPMAP
CLIST
OPT
TRUNC
FLOW
STATE
NAME
SYMDMP
TEST

CNDENSE=

CLIST

specifies that a condensed listing is to be produced. A condensed listing contains, in the procedure portion, the source card numbers and the first instruction generated for each verb.

NOCLIST

specifies that no condensed listing is to be produced.

Note: If CNDENSE=CLIST is specified, PROCMAP=PMAP cannot be specified.

COUNTL=

COUNT

specifies that object code is to be generated to produce verb execution summaries at the end of problem program execution. Each verb is identified by procedure-name and by statement number; the number of times it was also used is indicated. In addition, the percentage of verb execution for each verb with respect to the execution of all verbs is given. A summary of all verbs used in a program and the number of times they are executed is provided. COUNT implies VERB.

NOCOUNT

specifies that verb execution statistics will not be produced.

DATAMAP=

specifies the production of a listing of the data-names and their addresses relative either to load point for the Working-Storage Section or to the record addresses for the File or Linkage Sections.

DMAP

specifies that a listing is to be produced.

NODMAP

specifies that no listing is to be produced.

DUMP=

DUMP

specifies that the compiler will issue an ABEND for a D-level message condition.

NODUMP

specifies that the abend will not be issued, and that a D-level message will be produced.

ENDJOB=

specifies whether or not, at the end of each job, COBOL Library subroutines are to be called to delete modules, free main storage acquired through GETMAINs issued by the COBOL program or COBOL Library subroutines, close DCBs opened by subroutines, and free their associated buffers.

Specifying ENDJOB prevents fragmentation of main storage for programs executed on the system after the COBOL program has been executed. This option takes effect at a STOP RUN statement in any program and at a GOBACK statement in a main program only.

ENDJOB

specifies that the ENDJOB function is to be performed.

NOENDJOB

specifies that the ENDJOB function is not to be performed.

FLOW=

specifies whether a formatted trace of a variable number of procedures executed before an abnormal termination is to be listed.

FLOW

specifies that the trace is to be produced.

NOFLOW

specifies that no trace is to be produced.

FLOWSIZ=

specifies the default value for the number of procedures in the formatted trace when the flow trace option is used. At installation time, this number can be specified even if the flow trace option is not requested as a default.

nn

a 1- or 2-digit integer from 1 to 99. If FLOWSIZ=nn is not specified, nn is set to 99.

LIBRARY=

specifies whether the Library Management facility, which allows a single copy of a library subroutine to be shared by all COBOL programs in the same or different partitions or regions, is to be used.

RESIDENT

specifies that library management is in effect.

NORESIDENT

specifies that library management is not in effect.

Note: RESIDENT is always in effect when the DYNAM option is used.

LINECNT=

specifies the number of lines to be printed on each page of the COBOL compiler output listing.

lines

is a 2-digit number from 10 to 99. If this parameter is omitted, a value of 60 is assumed.

LITCHAR=

specifies the character that is to be used to delineate literals. It is also to be used by the compiler to delineate figurative constants.

APOST

specifies that the apostrophe (') is to be used.

QUOTE

specifies that the double quote (") is to be used.

LSTCOMP=

LSTCOMP

specifies that, in addition to the listing and optional deck produced by the LSTONLY option, the source program is to be compiled.

NOLST

specifies that the lister feature is not to be used.

Note: When NOLST is in effect, the CDECK, FDECK, LCOL1, LCOL2, and L120 options are ignored. If NOLST and NOBATCH are in effect, the L132 option is ignored.

LSTDECK=

FDECK=

specifies that an updated source deck is to be produced. If the CDECK option is in effect, the updated source deck includes the updated and reformatted copy libraries.

Note: If NOLST is in effect, the FDECK option is ignored.

NOFDECK

specifies that an updated source deck is not to be produced.

LSTONLY=

LSTONLY

specifies that a listing of the lister feature reformatted source program is to be produced, but that the program is not to be compiled; if the FDECK option is in effect, the updated source deck will also be produced.

NOLST

specifies that the lister feature is not to be used.

Note: When NOLST is in effect, the CDECK, FDECK, LCOL1, LCOL2, and L120 options are ignored. If NOLST and NOBATCH are in effect, the L132 option is ignored.

LSTPCH=

CDECK

specifies that the updated and reformatted copy libraries are to be punched. If the FDECK option is in effect, the libraries are punched as part of the source deck. If the NOFDECK option is in effect, the libraries are punched as a separate deck.

Note: If NOLST or NOLIB is in effect, the CDECK option is ignored.

NOCDECK

specifies that the updated and reformatted copy libraries are not to be punched.

LSTPRC2=

LCOL1

specifies that the Procedure Division is to be listed in single-column format.

LCOL2

specifies that the Procedure Division is to be listed in double-column format.

Note: If NOLST is in effect, this option is ignored.

LST132=

L120

specifies that the print line is to be 120 characters long.

L132

specifies that the print line is to be 132 characters long.

Note: If NOLST and NOBATCH is in effect, this option is ignored.

LVL=

specifies whether the Federal Information Processing Standard (FIPS) flagger is to be activated, and at what level.

NOLVL

specifies no flagging.

A

specifies low level flagging.

B
specifies low intermediate flagging.

C
specifies high intermediate flagging.

D
specifies full standard flagging.

For a list of the statements flagged at each level, see IBM VS COBOL for OS/VS.

Notes:

1. If LVL=level is the SYSGEN default, it can be overridden during compile time with any parameter except NOLVL.
2. When LVL=level is in effect, the SYSUT6 data set must be specified during compile time.
3. When LVL=level is in effect, the LANGLVL specification establishes whether the 1972 or 1975 FIPS is to be used. LANGLVL(1) indicates 1972; LANGLVL(2) indicates 1975.

MSGLEV=
specifies the type of compilation error messages to be printed.

FLAGW
specifies that all warning and error messages are to be printed.

FLAGE
specifies that warning messages are not to be printed.

NAME=
specifies whether each program in a batch compilation is to be link-edited as a separate load module.

NAME
specifies separate load modules.

NONAME
specifies no separate load modules.

NEWADV=

ADV
specifies that records in files for which WRITE...ADVANCING has been coded need not have the first byte reserved for the control character.

NOADV
specifies that the first byte must be reserved for the control character.

NUM=
specifies the use of user-supplied line numbers recorded in the input instead of compiler-generated statement numbers wherever such numbers are listed.

NUM
specifies that user-supplied line numbers are to be used.

NONUM
specifies that user-supplied line numbers are not to be used.

OPT=

OPT
specifies that the output object module is to have optimized object code.

NOOPT
specifies that the output object module code is not to be optimized.

PROCMAP=
specifies the production of a listing of the instructions generated for each statement in the Procedure Division.

PMAP
specifies that the listing is to be produced.

NOPMAP
specifies that no listing is to be produced.

Note: If PROCMAP=PMAP is specified, CNDENSE=CLIST cannot be specified.

PUNCH=
specifies the production of a punched deck of the object program.

DECK
specifies that a punched deck is to be produced.

NODECK
specifies that a punched deck is not to be produced.

REFLIST=

XREF
specifies that an unsorted cross-reference listing is to be produced.

NOXREF
specifies that no cross-reference listing is to be produced.

Note: If REFLIST=XREF is specified, SREFLST=SXREF cannot be specified.

SEQCHK=
specifies the checking of the source program card sequence numbers.

SEQ
specifies that the source program card sequence numbers are to be checked.

NOSEQ
specifies that the source program card sequence numbers are not to be checked.

SIZE=
specifies the number of bytes of main storage available to the COBOL compiler.

size
is an integer from 131072 to 9999999. If this parameter is omitted, a value of 131072 is assumed. For further information on this parameter, see IBM OS/VS COBOL Compiler and Library Programmer's Guide.

SORLIST=
specifies the production of a listing of the COBOL source program.

SOURCE
specifies that the listing is to be produced.

NOSOURCE
specifies that the listing is not to be produced.

SPACE=
specifies the line spacing on the listing obtained when the SOURCE option is specified.

SPACE1
specifies single spacing.

SPACE2
specifies double spacing.

SPACE3
specifies triple spacing.

SREFLST=

SXREF
specifies that a sorted cross-reference listing is to be produced.

NOSXREF
specifies that a sorted cross-reference listing is not to be produced.

Note: If SREFLST=SXREF is specified, REFLIST=XREF cannot be specified.

STATE=
specifies whether the number of the statement and verb being executed at the time of an abnormal termination is to be listed. STATE should not be specified when SYMDMP is in effect.

STATE
specifies that the listing is to be produced.

NOSTATE
specifies that no listing is to be produced.

STD=
specifies whether the 1968 or 1974 American National Standard COBOL definition (as understood and interpreted by IBM) is to be used when compiling those source elements whose meaning has changed.

Generally speaking, the language supported by the Release 2 compiler includes all of that supported by Release 1. The Release 2 compiler accepts not only source programs written in the new (1974) language, but also programs written in the older (1968) language. However, the superset relationship between the new and the older languages is not absolute; there are a few exceptions—elements whose meaning has changed because of ANS redefinition. (These are discussed in "Section III—'Old' Language Extensions" of Appendix A of IBM VS COBOL for OS/VS.) It is only these few elements that are controlled by the LANGLVL option. Language elements whose meanings did not change from 1968 ANS to 1974 ANS (the vast majority of the language) are unaffected by the LANGLVL option, as are those language elements new in the 1974 ANS definition. The compiler accepts all such elements regardless of the LANGLVL specification.

LANGLVL(1)
tells the compiler to use the 1968 ANS standard for those source elements whose definition has changed. (This interpretation is the one used by Release 1 of the compiler.)

LANGLVL(2)
tells the compiler to use the 1974 ANS standard for those source elements whose definition has changed from ANS 1968.

SUPPRES=

NOSUPMAP
specifies that the object code listing, the object deck, and the load module are to be produced even if an E-level message is generated by the compiler.

SUPMAP
specifies that the above items are to be suppressed when an E-level message is generated.

SYMDMP=
specifies whether a formatted symbolic dump of specified data areas can be listed dynamically on the SYSPRINT data set at various points during program execution; in the event of abnormal termination, a formatted symbolic dump of all data areas is to be produced. When SYMDMP is in effect, OPT is always in effect; STATE and BATCH should not be specified.

SYMDMP
the formatted dumps are to be produced as required.

NOSYMDMP
the formatted dumps are not to be produced.

SYSLIB=
specifies whether or not the data set used for COPY and BASIS statements is to be opened during compilation.

LIB
specifies that the data set, which is defined by a SYSLIB DD statement, is to be opened.

NOLIB
specifies that the data set is not to be opened.

SYSOUT=
specifies that either SYSOUT or SYSOUx is to be the ddname of the file to be used for debug output and for data when SYSOUT is specified in a DISPLAY statement.

SYST
specifies that SYSOUT is to be the ddname of the file to be used for debug output and for data when SYSOUT is specified in a DISPLAY statement.

SYSx
specifies that SYSOUx, where x must be alphanumeric (that is 0 through 9 or A through Z except for T), is to be the ddname of the file to be used for debug output and for data when SYSOUT is specified in a DISPLAY statement.

TERM=
specifies the listing of progress and diagnostic messages and compiler statistics on the SYSTERM data set.

TERM
specifies that the messages and statistics are to be listed.

NOTERM
specifies that no messages or statistics are to be listed.

TEST=

specifies whether or not the IBM Program Product IBM OS COBOL Interactive Debug may be used at execution time.

TEST

specifies that Interactive Debug may be used.

NOTEST

specifies that Interactive Debug may not be used.

TRNCATE=

specifies the treatment of computational (binary) items during a move operation.

TRUNC

specifies that the item is moved to a receiving field according to the specification in the PICTURE clause.

NOTRUNC

specifies that the item is moved to a receiving field according to the size of the field in storage (halfword, fullword, and so on).

TYPERUN=

specifies the production of input to the linkage editor or loader from the program being compiled.

LOAD

specifies that the program is to be processed by the linkage editor or loader after compilation.

NOLOAD

specifies that the program is only to be compiled.

VERB=

specifies whether procedure and verb-names are to be listed with the associated code for PMAP or CLIST, and specifies that either procedure names or card numbers are to be used in READY TRACE output.

VERB

specifies that procedure and verb-names are to be listed with the associated code, and procedure-names are to be used for READY TRACE output.

NOVERB

specifies that only the statement number is to be listed for PMAP or that only the words VERB1, VERB2, and so on, are to be listed along with the statement number for CLIST. If there are no READY TRACE statements, or PMAP or CLIST is not in effect, NOVERB should be specified for efficient processing.

VRBREF=

VBREF

specifies that a cross reference of all verbs used in the program is to be produced. This option provides the programmer with a quick index to any verb used in the program.

Note: VRBREF=VBREF requires a specification of VERB=VERB and VRBSUM=VBSUM.

NOVBREF

specifies that a verb cross reference is not to be produced.

VRBSUM=

VBSUM
specifies that a brief summary of verbs used in the program is to be produced, along with a count of how often each verb is used. This option provides the user with a quick search for specific types of statements.

Note: VRBSUM=VBSUM requires a specification of VERB=VERB.

NOVBSUM
specifies that a verb summary is not to be produced.

ZWHENB=
specifies whether or not the compiler is to generate code to strip the sign from a signed external decimal field when comparing this field to an alphanumeric field.

ZWB
specifies that the sign of the signed external decimal field is to be stripped.

NOZWB
specifies that the sign of the signed external decimal field is not to be stripped.

If you receive a message describing any invalid or conflicting values on the VSCBL1 macro instruction, installation of the compiler cannot proceed. Respecify VSCBL1 with correct values and rerun the installation job.

PART 2. INSTALLATION PROCEDURES

This part contains an overview of the installation procedures you use to install the OS/VS COBOL product.

- **Installing in OS/VS1 or OS/VS2** on page 20

  Provides an overview of how to install in an OS/VS1 or OS/VS2 operating system.

- **Installing in VM/CMS** on page 22

  Provides an overview of how to install in a VM/CMS operating system.

INSTALLATION PROCEDURES

An outline of the requirements for installation and of the necessary steps is given in the following sections. For complete details of the steps for installing the Compiler and Library, see the OS/VS COBOL Program Directory.

Use the following outline of the steps for your planning. Use this outline plus the OS/VS COBOL Program Directory while you are installing this product.

Note: The modules of the OS/VS COBOL Release 2 Compiler and its associated subroutine library have the same names as those in previous OS/VS and OS COBOL Compilers and Libraries. If the new Compiler and Library are added to a system containing an earlier version, the earlier version will be overlaid, unless you take steps to ensure its preservation.

INSTALLING IN OS/VS1 OR OS/VS2

This section describes OS/VS COBOL Compiler and Library installation under OS/VS1 or OS/VS2. The steps that follow present a brief overview of the installation process.

For detailed information on the installation of the compiler, and on the installation of the library, see the OS/VS COBOL Program Directory.

INSTALLATION PROCESS—COMPILER AND LIBRARY—OS/VS

Before starting the installation process, be sure you have specified your COBOL options by coding the VSCBL1 macro. This step is discussed in "Specifying the Compiler Default Options" on page 6, and in the OS/VS COBOL Program Directory.

To install the OS/VS COBOL Compiler and Library using SMP4:

1. Use the sample copy job in the program directory to copy the System Modification Program (SMP) procedures from the PID tape.

2. Code the VSCBL1 macro to specify the default compiler options. Place them in the SYS1.PPOPTION library.

3. Use SMP4 to install the Compiler and/or Library, using the following procedures:

   a. ALLCPROC, to allocate space for the required data sets

   b. INITPROC, to initialize SMP data sets

   c. DISTPROC, to allocate space for the distribution data sets

   d. ACCCOB, to invoke SMP4 to access the required data sets

4. Use the SMP INSTALL job next. It uses the preceding procedures in step 3 to install the program product.

5. Optionally, run one or both sample programs to verify that installation is complete.

6. Use the SMP4 ACCEPT procedures to accept the product into the distribution libraries.

For complete details on the installation process under OS/VS, see the OS/VS COBOL Program Directory.

Verifying Success—OS/VS Compiler and Library Installation

Two sample OS/VS COBOL programs, TESTRUN and ERRMSG, are available to verify the success of the Compiler and Library installation. The sample programs are placed in SYS1.SAMPLIB during the installation process. The cataloged procedure COBUCLG in SYS1.PROCLIB can be used to compile, link-edit, and execute each sample program.

INSTALLATION PROCESS—LIBRARY ONLY—OS/VS

To only install the OS/VS COBOL Library using SMP4:

1. Use the sample copy job in the program directory to copy the SMP procedures from the PID tape.

2. Use SMP4 to install the Library using the following procedures:

   a. ALLCPROC, to allocate space for the required data sets

   b. INITPROC, to initialize SMP data sets

   c. DISTPROC, to allocate space for the distribution data sets

   d. ACCCOB, to invoke SMP4 to access the required data sets

3. Use the SMP INSTALL job next. It uses the preceding procedures in step 2 to install the program product.

4. Use the SMP4 ACCEPT procedures to accept the product into the distribution libraries.

For complete details on the installation process under OS/VS, see the OS/VS COBOL Program Directory.

INSTALLING IN VM/CMS

This section describes OS/VS COBOL installation under CMS.

INSTALLATION PROCESS—COMPILER AND LIBRARY—CMS

This section describes how to install the OS/VS COBOL Compiler and Library in the CMS environment.

> **Note:** Follow the procedure carefully. Do not interrupt the procedure or press the ENTER key unless so instructed.
>
> System responses in the following examples are shown in **boldface** type.

1. LOGON to the CMS environment.

2. Define the COBOL compiler default option for your installation by typing-in:

```
edit ikfcbl01 assemble
[Press the ENTER key].

NEW FILE
EDIT:

input
[Press the ENTER key].

INPUT

vscbl1
[Press the ENTER key].
.
.
.

For further details on the VSCBL1 macro instruction, see
"Specifying the Compiler Default Options" on page 6, or the
OS/VS COBOL Program Directory.
.
.
.

end
[Press the ENTER key].

EDIT
file
[Press the ENTER key].

R;
```

3. Mount the installation tape at virtual address 181. For details on mounting tapes, see IBM Virtual Machine Facility/370, Operator's Guide.

4. Move the Compiler and Library modules from the distribution tape into your system. You may use the following example of how to do this, but check the OS/VS COBOL Program Directory for CMS for current changes to the example.

Enter the following commands:

```
tape rew
[Press the ENTER key].

R;

tape load
[Press the ENTER key].

R;

exec cmscobol install
.
.
.
(System messages concerning the status of the files to be
created are printed at the terminal.)
.
.
.
INSTALL/REGEN COMPLETE
R;
```

Installation is now complete.

Replacing Compiler and Library Routines—CMS

To replace modules in the compiler or library:

1. Use the procedure listed in step 2 on page 22 to define the compiler default options.

2. Determine if CMSCOBOL EXEC is still available from installation time. If it is still available, go to step 3. If it is no longer available, mount the installation tape at virtual address 181. For details on mounting tapes, see *IBM Virtual Machine Facility/370, Operator's Guide*.

   Move the routines from the distribution tape into your system. You may use the following example of how to do this, but check the OS/VS COBOL Program Directory for current changes to the example.

   Enter the following commands:

   ```
   tape rew
   [Press the ENTER key].

   R;

   tape load
   [Press the ENTER key].

   R;
   ```

3. Replace the IBM-supplied routine.

   Names of replacement modules are listed in the OS/VS COBOL Program Directory. Enter the file name of the first module being replaced.

If the "filename" entered indicates that a compiler or library module should be replaced, the system responds by asking for the TXTLIB filename to which the replacement module will be added. Enter the correct TXTLIB filename. If the "filename" entered indicates that the VSCBL1 macro instruction is to be replaced, the system responds by asking for the MACLIB filename to which the macro instruction is to be added. Also, when replacing the VSCBL1 macro instruction, the file IKFCBL01 ASSEMBLE must have been created previously.

```
exec cmscobol ptf
"filename" [Press the ENTER key].

YOU MAY REPLACE COMPILER AND/OR LIBRARY ROUTINES.
SPECIFY THE FIRST OR NEXT COMPILER ROUTINE FILENAME.
IF FINISHED, PRESS THE ENTER KEY.

"filename" [Press the ENTER key].
```

If "filename" entered is VSCBL1:

```
SPECIFY THE TXTLIB NAME FOR THE NEW COMPILER MACRO TO BE
WRITTEN TO.
```

If "filename" entered is a compiler or library object file:

```
SPECIFY THE TXTLIB NAME THAT THE NEW  COMPILER ROUTINES
                                        LIBRARY
ARE TO BE WRITTEN TO.
BEWARE, THE TXTLIB CANNOT ALREADY CONTAIN A ROUTINE
BEING REPLACED.
"filename"        or
[ENTER]
```

Enter the "filename" of the next module to be replaced. This "filename" will be added to the same TXTLIB filename indicated in the previous response. When all modules have been added, enter a null line by pressing the ENTER key. (Compiler and Library routines are put into separate TXTLIBs.)

If none of the Compiler routines have been replaced, including the VSCBL1 macro instruction, the following occurs:

```
NO COMPILER MODULES REPLACED
INSTALL/REGEN COMPLETE
R;
```

If any of the Compiler routines have been replaced, including the VSCBL1 macro instruction, the following sequence occurs:

```
REGENING THE COMPILER CMS OVERLAY MODULES
IF "COBOLVS TXTLIB" THAT WAS CREATED AT INSTALL TIME
IS NOT AVAIL.  BY THAT NAME SPECIFY ITS CORRECT NAME
ELSE PRESS THE ENTER KEY.

[ENTER]
"txtlib filename"

TXTLIB "filename1" AND "filename2" ARE BEING USED.
WHEN REQUESTED, SPECIFY THE EXECUTION TIME MODE LETTER
OF THE DISK WHICH WILL CONTAIN THE COMPILER MODULES.
(THE VALID MODES ARE THE LETTERS A THRU Z.  THE SYSTEM
WILL SEARCH THE COBOL AUXILIARY DIRECTORY AT COMPILATION
TIME TO FIND THE LETTER THAT YOU ARE ABOUT TO SPECIFY,
AND WILL GET THE COBOL COMPILER MODULES FROM THAT DISK.)
INSTALL/REGEN COMPLETE

R;
```

"Filename1" is the txtlib filename specified when replacing Compiler routines. "Filename2" is the txtlib filename COBOLVS created at installation time or its equivalent. If the VSCBL1 macro instruction was replaced without any other Compiler routines being replaced, the system response is:

```
TXTLIB "filename1" IS BEING USED. ("Filename1" is
COBOLVS or its equivalent.)
```

The replacement procedure is then complete.

Regenerating the Compiler Overlay Structure—CMS

If the Compiler CMS overlay structure is to be regenerated with no modules being replaced, enter the following at the terminal:

```
exec cmscobol regen
[Press the ENTER key].

REGENING THE COMPILER CMS OVERLAY MODULES.
IF "COBOLVS TXTLIB" THAT WAS CREATED AT INSTALL
TIME IS NOT AVAIL. BY THAT NAME,
SPECIFY ITS CORRECT NAME, ELSE PRESS THE ENTER KEY.

[Press the ENTER key].
        or
"txtlib filename"

TXTLIB "filename" IS BEING USED
INSTALL/REGEN COMPLETE
R;
```

Regeneration of the Compiler overlay structure is then complete.

Messages and Codes—CMS

The installation procedures issue warning and diagnostic messages during installation. The identification codes and appropriate user responses follow.

Code: 1

Message: **INCOR. ARGUMENT "argument"**

Response: Reenter the "exec cmscobol" command with the correct argument: "install", "ptf", or "regen".

Code 2

Message: **"filename filetype" ALREADY EXIST . . . RENAME OR ERASE IT AND TRY AGAIN.**

Response: A file with a reserved file ID was found on an accessed disk. Rename or erase it and try again.

Code 3

Message: **ERROR FROM TAPE UNIT 181**

Response: Input/output error in positioning tape. Reissue the command.

Code 4–14

Message: **INTERNAL PROCESS ERROR**

Response: Contact your IBM installation representative.

Testing the Success of the Installation Procedure—CMS

The installation procedure provides a sample program on the user's A-disk. The sample program is called TESTRUN COBOL A1. To test the success of the installation, follow this terminal session:

```
cobol testrun  (dmap pmap sxref quote)
[Press the ENTER key].

REL. 2.4 OS/VS COBOL IN PROGRESS
R;

global txtlib coblibvs
[Press the ENTER key].

R;

load testrun
[Press the ENTER key].

R;

start
[Press the ENTER key].

R;
```

The output from the sample program consists of:

- Listing at the system printer of heading (including date and level), source program, map of the Data Division, map of the Procedure Division, module map, and alphabetically ordered cross-reference list.
- Output generated by the TESTRUN program at the terminal.

A listing of the TESTRUN sample program and the output produced by the program will be found in IBM OS/VS COBOL Compiler and Library Programmer's Guide.

Loading the Optional Machine-Readable Materials—CMS

To load the OS/VS COBOL source modules and generate the macro library under CMS, the OPTMAT procedure is used. OPTMAT loads the source files and also generates a macro library from the macro files on the tape. The CMS compile-time interface (DMSCOB), compiler directory (DMSCBD), and execution-time interface (DMSILB) are also loaded onto the disk.

To load the optional materials, you may use the following example, but check the OS/VS COBOL Program Directory for CMS for current changes to the example.

1. Mount the optional machine-readable material tape at virtual address 181. For details on mounting tapes, see IBM Virtual Machine Facility/370, Operator's Guide.

   Move the optional machine-readable materials from the distribution tape into your system. You may use the following example of how to do this, but check the OS/VS COBOL Program Directory for current changes to the example. See IBM Virtual Machine Facility/370, Operator's Guide.

2. Enter the following commands:

```
tape rew
[Press the ENTER key]

R;

tape fsf 3
[Press the ENTER key]

R;

tappds * exec a1 (update coll)
exec optmat
.
.
.

(The list of modules and macros copied from the optional
materials tape is printed at the terminal.)
.
.
.
LOAD COMPLETE
R;
```

At this point, the source modules are loaded and the macro library is generated.

MESSAGES AND CODES FOR OPTMAP PROCEDURE: The OPTMAT EXEC procedure issues warning and diagnostic messages to the terminal user. These messages, together with their identifying codes and appropriate user responses follow.

Code: 2

Message: **ERROR BUILDING "filename" MACLIB**

Response: Check the format of the macro files on disk for valid MACRO, MEND, and prototype records.

Code: 3

Message: **ERROR FROM TAPE UNIT 181**

Response: Input/output error in repositioning the tape. Reissue the command.

Code: All other codes

Message: **INTERNAL PROCESSING ERROR**

Response: Contact your IBM installation representative.

INSTALLATION PROCESS—LIBRARY ONLY—CMS

This section describes how to install the IBM OS/VS COBOL Library in the CMS environment.

To install the Library you may use the following example, but check the OS/VS COBOL Program Directory for CMS for current changes to the example.

1. Mount the Library-only installation tape at virtual address 181. For details on how to mount tapes, see IBM Virtual Machine Facility/370 Operator's Guide.

   Move the Library modules from the distribution tape into your system. You may use the following example of how to do this, but check the OS/VS COBOL Program Directory for current changes to the example.

2. Enter the following commands:

```
tape rew
[Press the ENTER key].

R;

tape load
[Press the ENTER key].

R;
exec libonly
[Press the ENTER key].

LIBRARY INSTALL COMPLETE
R;
```

At this point, installation of the Library, alone, is complete.

Messages and Codes for Library-Only Procedure

The installation procedure for the subroutine Library alone, issues warning and diagnostic messages during installation. These messages, together with the identification codes and appropriate user responses, follow.

Code: 1

Message: "COBLIBVS TXTLIB" ALREADY EXISTS; RENAME OR ERASE IT AND TRY AGAIN.

Response: A file with a reserved fileid was found on the accessed disk. Rename or erase it and retry the command.

Code: 3

Message: ERROR FROM TAPE UNIT 181

Response: Input/output error in repositioning the tape. Reissue the command.

Code: All other codes

Message: INTERNAL PROCESSING ERROR

Response: Contact your IBM installation representative.

PART 3. REFERENCE INFORMATION

APPENDIX A. PUBLICATIONS

DEVICE PUBLICATIONS

OS/VS COBOL supports those devices that are not apparent through the system access methods. For details on device capacities and minimum and maximum block size values, consult a publication that contains data for the I/O device you are using.

SUGGESTED PUBLICATIONS

OS/VS1

OS/VS1 Data Management Services Guide, GC26-3874

OS/VS1 Access Method Services, GC26-3840

OS/VS1 JCL Services, GC24-5100

OS/VS1 JCL Reference, GC24-5099

OS/VS2

OS/VS2 Data Management Services Guide, GC26-3875

OS/VS2 Access Method Services, GC26-3841

OS/VS2 MVS JCL, GC28-0692

VM/370

IBM Virtual Machine Facility/370: CP Command Reference for General Users, GC20-1820

VM/SP

IBM Virtual Machine/System Product: CP Command Reference for General Users, SC19-6211

APPENDIX B. LIBRARY MODULES

The following lists show the module name for each module in the COBOL object-time subroutine library.

Each module has an associated system status indicator (SSI). The SSI consists of 8 hexadecimal digits: The first 2 are always E2; the next 2 are the release level and the modification level; the last 4 are zeros.

OS/VS COBOL SUBROUTINE LIBRARY MODULES

The following summary lists all modules in the OS/VS COBOL subroutine library.

Module	Alias	Characteristic*
ILBOABN	ILBOABN0	R
ILBOACP	ILBOACP0	R
ILBOACS	ILBOACS0,ILBOACS1	R
ILBOANE	ILBOANE0	R
ILBOANF	ILBOANF0	R
ILBOATB	ILBOATB0	R
ILBOBEG	ILBOBEG0	R
ILBOBID	ILBOBID1,ILBOBID2,ILBOBID0	R
ILBOBIE	ILBOBIE1,ILBOBIE2,ILBOBIE0	R
ILBOBII	ILBOBII1,ILBOBII2,ILBOBII0	R
ILBOBUG	ILBOBUG0,ILBOBUG1	R
ILBOCHN	ILBOCHN0	R
ILBOCJS		NU
ILBOCKP	ILBOCKP1,ILBOCKP0	R
ILBOCLS	ILBOCLS0	R
ILBOCMM	ILBOCMM0	R
ILBOCOM	ILBOCOM0	N
ILBOCT1	ILBOCT10	R
ILBOCVB	ILBOCVB0,ILBOCVB1	R
ILBODBE	ILBODBE0	R
ILBODBG	ILBODBG0,ILBODBG1,ILBODBG2 ILBODBG3,ILBODBG4,ILBODBG5,ILBODBG6,ILBODBG7	R
ILBODCI	ILBODCI1,ILBODCI0	R
ILBODSP	ILBODSP0	R
ILBODSS	ILBODSS0	N
ILBODTE	ILBODTE1,ILBODTE0,ILBODTE2 ILBODTE3,ILBODTE4	R
ILBOD01	ILBOD010	R
ILBOD10	ILBOD100	R
ILBOD11	ILBOD110	R
ILBOD12	ILBOD120	R
ILBOD13	ILBOD130	R
ILBOD14	ILBOD140	R
ILBOD20	ILBOD200	R
ILBOD21	ILBOD210	R
ILBOD22	ILBOD220	R
ILBOD23	ILBOD230	R
ILBOD24	ILBOD240	R
ILBOD25	ILBOD250	R
ILBOD26	ILBOD260	R
ILBOEFL	ILBOEFL1,ILBOEFL2,ILBOEFL0	R
ILBOERR	ILBOERR1,ILBOERR2,ILBOERR0,ILBOERR3 ILBOERR4,ILBOERR5,ILBOERR6,ILBOERR7	R
ILBOETB	ILBOETB0	R
ILBOEXT	ILBOEXT0,ILBOEXT1	R
ILBOFLW	ILBOFLW1,ILBOFLW2,ILBOFLW0	R
ILBOFPW	ILBOFPW0	R
ILBOGDO	ILBOGDO0,ILBOGDO1,ILBOGDO2	R

*R indicates the module is reentrant, N indicates it is nonreentrant (but reusable), and NU indicates it is both nonreentrant and nonreusable.

Module	Alias	Characteristic*
ILBOGPW	ILBOGPW0	R
ILBOIDB	ILBOIDB1,ILBOIDB0	R
ILBOIDR	ILBOIDR0	R
ILBOIDT	ILBOIDT0	R
ILBOIFB	ILBOIFB1,ILBOIFB2,ILBOIFB0	R
ILBOIFD	ILBOIFD1,ILBOIFD0	R
ILBOINS	ILBOINS0	R
ILBOINT	ILBOINT0	R
ILBOITB	ILBOITB0	R
ILBOIVL	ILBOIVL0	R
ILBOLBL	ILBOLBL0,ILBOLBL1,ILBOLBL2,ILBOLBL3	R
ILBOMRG	ILBOMRG0	R
ILBOMSC	ILBOMSC0	R
ILBOMSG	ILBOMSG0	R
ILBONBL	ILBONBL0	R
ILBONED	ILBONED0	R
ILBONTR	ILBONTR0	R
ILBOOCR	ILBOOCR0	N
ILBOPRM	ILBOPRM0	R
ILBOPTV	ILBOPTV1,ILBOPTV2,ILBOPTV0	R
ILBOQIO	ILBOQIO0,ILBOQIO1	R
ILBOQSS	ILBOQSS0,ILBOQSS1	R
ILBOQSU	ILBOQSU0,BLDQS	NU
ILBOREC	ILBOREC0	R
ILBORNT	ILBORNT0	R
ILBOSAM	ILBOSAM0	R
ILBOSCD	ILBOSCD0	R
ILBOSCH	ILBOSCH0	R
ILBOSDB	ILBOSDB0	R
ILBOSGM	ILBOSGM1,ILBOSGM0,ILBOSGM2,ILBOSGM3	R
ILBOSMG	ILBOSMG0	R
ILBOSMV	ILBOSMV0	R
ILBOSND	ILBOSND0,ILBOSND1,ILBOSND2	R
ILBOSNT	ILBOSNT0	R
ILBOSPA	ILBOSPA1,ILBOSPA0,ILBOSPA2	R
ILBOSPI	ILBOSPI0	R
ILBOSPN	ILBOSPN0,ILBOSAMR	R
ILBOSQA	ILBOSQA1,ILBOSQA2,ILBOSQA0	NU
ILBOSRT	ILBOSRT0	R
ILBOSRV	ILBOSR,ILBOSRV0,ILBOSRV1,ILBOSR3, ILBOSR5,ILBOST,ILBOSTP1,ILBOSTP0	N
ILBOSSN	ILBOSSN1,ILBOSSN0	R
ILBOSTG	ILBOSTG0	R
ILBOSTI	ILBOSTI0	R
ILBOSTN	ILBOSTN0	
ILBOSTR	ILBOSTR0	R
ILBOSTT	ILBOSTT0	R
ILBOSYN	ILBOSYN0,ILBOSYN1,ILBOSYN2, ILBOSYN3,ILBOSYN4,ILBOSYN5	R
ILBOTC0	ILBOTC00	R
ILBOTC2	ILBOTC20,ILBOTC21	R
ILBOTC3	ILBOTC30	R
ILBOTEF	ILBOTEF1,ILBOTEF2,ILBOTEF3 ILBOTEF0	R
ILBOTRN	ILBOTRN0	R
ILBOUST	ILBOUST0	R
ILBOUTB	ILBOUTB0	R
ILBOVCO	ILBOVCO0	R
ILBOVIO	ILBOVIO0	R
ILBOVMO	ILBOVMO1,ILBOVMO0	R
ILBOVOC	ILBOVOC0,ILBOVOC1	R
ILBOVTR	ILBOVTR0	R
ILBOWAT	ILBOWAT0	R
ILBOWTB	ILBOWTB0	R
ILBOXDI	ILBOXDI0	R
ILBOXMU	ILBOXMU0	R
ILBOXPR	ILBOXPR0	R

*R indicates the module is reentrant, N indicates it is nonreentrant (but reusable), and NU indicates it is both nonreentrant and nonreusable.

INDEX

Reader's Comment Form

IBM OS/VS COBOL Compiler and Library
Installation Reference Material
SC28-6481-2

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

List TNLs here:

If you have applied any technical newsletters (TNLs) to this book, please list them here:

Last TNL ____________________

Previous TNL ________________

Previous TNL ________________

Fold on two lines, tape, and mail. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.) Thank you for your cooperation.

Note: Staples can cause problems [illegible] automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

SC28-6481-2

Reader's Comment Form

Fold and tape | Please do not staple | Fold and tape

IBM Corporation
P.O. Box 50020
Programming Publishing
San Jose, California 95150

Fold and tape | Please do not staple | Fold and tape

IBM®

SC28-6481-2

IBM OS/VS COBOL Compiler and Library Installation Reference Material (File No. S370-24) Printed in U.S.A. SC28-6481-2