SC28-6479-3

File No. S370-24

Program Product

IBM DOS/VS COBOL Compiler and Library Installation Reference Material

Program Numbers 5746-CB1
5746-LM4

Release 3

PREFACE

This publication contains the system-related information needed to install the IBM DOS/VS COBOL Compiler and its associated COBOL Object-time Subroutine Library. These program products operate under control of the IBM DOS/VSE System and the Conversational Monitor System (CMS) component of VM/SP (Virtual Machine Facility/System Product).

This publication is designed to be used in conjunction with the DOS/VS COBOL Program Directory.

Program Product Installation is directed to the system programmer or planner responsible for the planning and/or implementation of system generation and maintenance. It contains information corresponding to that found in DOS/VSE System Generation. The operating system requirements, minimum machine configuration, work file requirements, and IOCS modules required for the DOS/VS compiler are described, as is the procedure for installing DOS/VS COBOL under DOS/VSE and VM/SP. This section also contains storage requirements for the compiler and the library.

CMS messages produced for DOS/VS COBOL are described in the "Using DOS COBOL Under CMS" section of IBM CMS User's Guide for COBOL, SC28-6469.

ADDITIONAL PUBLICATIONS

Within the text, references are made to the following publications:

DOS/VSE System Management Guide, GC33-5371

DOS/VSE System Generation, GC33-5377

DOS/VSE System Control Statement, GC33-5376

DOS/VSE Maintain System History Program (MSHP) User's Guide, GC33-6060

DOS/VS COBOL Compiler and Library Programmer's Guide, SC28-6478

Virtual Machine Facility/SP Planning and System Generation Guide, SC19-6203

Virtual Machine Facility/SP CMS User's Guide, SC19-6210

DOS/VS Sort/Merge Programmer's Guide, SC33-4044

DOS/VS Sort/Merge Installation Reference, SC33-4045

Fourth Edition (May 1981)

This is a major revision of, and makes obsolete, SC28-6479-2, and its technical newsletters SN20-9181, SN20-9233, and SN20-9293.

This edition applies to Release 3 of the IBM DOS/VS Compiler and Library, Program Products 5746-CB1 and 5746-LM4, respectively, 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. Because the technical changes in this edition are extensive and difficult to localize, they are not marked by vertical bars in the left margin.

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.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

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 reader's 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 any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Summary of Amendments — Number 6

Form of Publication: Revision

Maintain System History Program (MSHP)

New: Installation Procedures

DOS/VS COBOL Release 3 is installed by the Maintain System History Program (MSHSP). The installation procedures have been rewritten to reflect the change.

Summary of Amendments — Number 5

Form of Publication: TNL SN20-9293

Support of Fixed Block Devices

New: Programming Feature

Support for fixed block devices is provided under DOS/VSE with VSE/Advanced Function, Release 1.

Summary of Amendments — Number 4

Form of Publication: TNL SN20-9233

Support of Additional Mass Storage Devices

New: Programming Feature

Support has been added for the 3330-11 Disk Storage and 3350 Direct Access Storage devices.

Summary of Amendments — Number 3

Date of Publication: December 3, 1976
Form of Publication: TNL SN20-9181 to SC28-6479-2

IBM DOS/VS COBOL

Maintenance: Documentation

- Minor technical changes and additions have been made to the text.

CONTENTS

FIGURES

The IBM DOS/VS COBOL Compiler ("the compiler") and Library is a Program Product that operates under control of the IBM DOS/VSE system or the Conversational Monitor System (CMS) component of VM/SP. It contains the COBOL compiler, as well as the COBOL subroutines that, when required, are combined by the Linkage Editor with the object program produced by the compiler. It also contains a set of transient routines that can be fetched dynamically into the user's storage area during program execution. It also contains a special set of COBOL statements that can be used to perform 3886 processing. It accepts as input any program written in DOS/VS COBOL.

The DOS/VS COBOL Object-time Subroutine Library ("the library") associated with the DOS/VS compiler is also available as a separate Program Product. It contains COBOL subroutines that, when required, are combined by the Linkage Editor with the object program produced by the DOS/VS compiler. The subroutine library also contains a set of transient routines that can be fetched dynamically into the transient area or loaded dynamically into the user's storage area during program execution.

Link-edited programs that have been compiled by the DOS/VS COBOL compiler (5746-CB1) all require the presence of the DOS/VS COBOL Library at execution time. Note that a separate library may not be needed. The library packaged with the compiler can be used if the programs are executing on the same processor as the one on which they are compiled or if they are executing under CMS; otherwise, there must be a separate library (5746-LM4) for each processor on which the programs are executing.

The CMS installation must order and install DOS/VSE (Advanced Function Release 3 or later). If the DOS/VS COBOL programs access any VSAM data sets, VSE/VSAM must also be installed. If the DOS/VS COBOL programs use SORT or MERGE, DOS/VSE Sort/Merge must be installed. Then, the CMS installation must install the DOS/VS COBOL compiler and library. The compiler must be link-edited in the background partition of DOS/VS, and all of the compiler's relocatable modules must be link-edited using the ACTION REL CLEAR linkage editor control statement. The compiler and library may reside on the DOS/VSE system pack or on private libraries.

When the DOS/VS COBOL compiler is executed under CMS, it must be executed in the CMS/DOS environment. Before invoking the DOS/VS COBOL compiler or executing DOS/VS COBOL programs under CMS, be sure that the CMS/DOS environment has been previously generated into your CMS system and is active. Also, when the DOS/VS COBOL compiler and DOS/VS COBOL programs are executed under CMS, the DOS/VSE system pack or private libraries must be available to CMS.

This section contains the information you will need to add the IBM DOS/VS COBOL Compiler and its associated COBOL Object-time Subroutine library to your DOS/VSE System.

The first part of this section, "System Requirements," describes the operating system requirements, minimum machine configuration, work file requirements, and LIOCS modules required by the compiler. The second part, "Installation Procedure," gives instructions for adding the compiler and the library to the system. The third part, "Storage Requirements," describes the amount of space required by the compiler and the library.

SYSTEM REQUIREMENTS

For DOS/VS, the compiler and the library operate under control of the IBM Disk Operating System (DOS/VSE). A DOS/VSE Advanced Function Release 3 system is the minimum level required. The minimum machine configuration and operating system requirements to support the compiler and library are described in this section. Execution-time considerations and special requirements for use of the Sort feature and the Symbolic Debug feature are also discussed.

For CMS, the compiler and library are controlled by the CMS/DOS environment. A DOS/VSE system pack or DOS/VSE private libraries must be available for CMS/DOS use. CMS/DOS requires a DOS/VSE Advanced Function Release 3 system as a minimum level. The minimum virtual machine configuration and operating system requirements to support the compiler and library under CMS are described in this section. Other execution-time considerations are also discussed in this section.

OPERATING SYSTEM REQUIREMENTS

If the SORT or MERGE verb is used in the source program under DOS/VSE, the DOS/VSE Sort/Merge Program Product, Program Number 5746-SM2, is required. For further information on this product, see DOS/VS SORT/MERGE Programmer's Guide and DOS/VS SORT/MERGE Installation Reference Material.

During link-editing, modules produced by the compiler require subroutines from the library. In addition, subroutines from the library may be required during execution. Therefore, the library must also be part of the system used to control program execution.

CMS requires access to the DOS/VSE System pack or the private libraries that contain the DOS/VS COBOL compiler or subroutines. To make the system pack available at virtual address 195 as your C-disk, specify the filemode when the CMS/DOS environment is invoked:

```
ACCESS 195      C
SET DOS ON      C
```

To make private libraries at address 193 available, access the disk specifying some CMS mode letter such as D, and issue ASSGN and DLBL commands after the CMS/DOS environment is invoked:

```
ACCESS 193      D
SET DOS ON
ASSGN  SYSCLB   D
DLBL   IJSYSCL  D  DSN ? (SYSCLB)
ASSGN  SYSRLB   D
DLBL   JJSYSKL  D  DSN ? (SYSRLB)
```

MACHINE CONFIGURATION FOR DOS/VSE

The minimum DOS/VSE machine configuration required to use the DOS/VS COBOL compiler and library is:

1. A System/370, 303X or 4300 model supported by DOS/VSE. At least 128K bytes must be available to the compiler.

2. Six work files.

   The system logical unit SYSLNK must be assigned to a single area (extent) on a 2311, 2314, 2319, 3330, 3340, 3350, 3375, or fixed block mass storage device.

   Five programmer logical units (SYS001 through SYS005) must reside on 2400, 3410, or 3420 tape units, or on 2311, 2314, 2319, 3330, 3340, 3350, 3375, or fixed block mass storage devices. (Two programmer logical units, as well as the operating system, must reside on a mass storage device.)

   If the four remaining logical units reside on tape, there must be a separate tape unit for each data set. If they reside on a mass storage device, there must be enough space on that device. See "Work File Requirements" for further information.

   Workfile assignments must be made as follows:

   ```
   SYS001 -- disk unit
   SYS002 -- disk or tape unit
   SYS003 -- disk or tape unit
   SYS004 -- disk or tape unit
   SYS005 -- disk or tape unit (required when Symbolic Debug is
             requested at compile time)
   SYS006 -- disk unit (required for FIPS)
   ```

   Note that SYSLNK need not be assigned at compile time unless the CATAL or LINK option is in effect.

   The filenames for SYSLNK and SYS001 through SYS006 on the TLBL or DLBL statements are IJSYSLN, IJSYS01, IJSYS02, IJSYS03, IJSYS04, IJSYS05, and IJSYS06, respectively.

3. A device for direct operator communication.

4. A device, such as a card reader, for the job input stream.

5. A device, such as a printer or tape unit, for system output files.

6. The floating-point arithmetic feature, if floating point literals or calculations are used.

Note: All devices currently supported by previous IBM DOS COBOL compilers are supported by the IBM DOS/VS COBOL Compiler. These include: 3504/3505 (with OMR), 3525 (with RCE and combined function processing), 3410/3420 Tapes, 3881 Optical Mark Reader, and 3330 Disk. The DOS/VS compiler also supports the 5425 Multifunction Card Unit, 2560 Multifunction Card Machine, 3203 and 5203 Advanced Printers, 3886 Optical Character Reader, 3340 Disk Facility, 3344 Direct Access Storage, 3350 Direct Access Storage in 3330-1 compatibility mode, and 3540 Diskette Input/Output Unit.

Execution-Time Consideration

The amount of virtual storage must be sufficient to accommodate at least:

- The selected control program
- Support for the file processing techniques used
- The load module to be executed

If symbolic debugging is requested, an additional work file, SYS005, must be assigned at compile time. This file must also be assigned at execution time if symbolic debugging control cards are present, though not necessarily to SYS005.

If execution statistics, VSAM, or the 3886 Optical Character Reader feature are requested, additional virtual storage must be available to accommodate its dynamic space requirements.

Use of the symbolic debugging, flow trace, statement number, execution statistics, option or the 3886 Optical Character Reader feature requires additional storage during execution. See IBM DOS/VS COBOL Compiler and Library Programmer's Guide for further information.

Sort/Merge Feature Considerations

The DOS/VSE Sort/Merge Program Product, Program Number 5746-SM2, must be executed under control of DOS/VSE. It requires the following minimum machine configuration.

1. The DOS/VS Sort/Merge Program Product uses 40K bytes; an additional 6K bytes are needed for DOS/VSE and for user-written routines.

   Note: Performance often increases significantly if 50K bytes is available for operation of the Sort/Merge program. At the 100K-byte level, the performance could be even higher.

2. Standard instruction set.

3. At least one 2314, 2319, 3330, 3333, 3340, or 3350 work file. (System residence requirements may necessitate having an additional disk storage unit for sorting.)

4. One IBM 1403, 1443, 3203, 5203, or 3211 Printer, or one IBM operator communication device (for example, 3215).

5. One IBM 1442, 2501, 2520, 2540, 3505, 3525, or 2560 Card Reader, or one IBM 2400 or 3400 Series Magnetic Tape Unit (7- or 9-track) assigned to SYSIPT and SYSRDR.

6. Three IBM 2400 or 3400 Series Magnetic Tape Units for work files when tape units are to be used for intermediate storage. For a merge operation, no work units are required.

Symbolic Debug Considerations

To use the Symbolic Debug feature of the DOS/VSE compiler, a fifth work file (SYS005) is required at compile time. When symbolic debugging is requested, the compiler records in SYS005 the information needed to produce a symbolic formatted dump if the program terminates abnormally or if dynamic dumps are requested. Thus, the file assigned to SYS005 at compile time must also be assigned at execution time. This file need not necessarily be assigned to SYS005, but it must be assigned in the partition in which the program is executed.

SYS005 can be allocated with the other system work files (SYS001, SYS002, SYS003, SYS004) at system generation time. Each COBOL source program requesting symbolic debugging then uses the same work file. However, unique symbolic debugging work file assignments are required when several COBOL programs are to be compiled in the same job. If there is only one SYS005 allocated for use by all COBOL programs, information recorded on the file during compilation of one program is overlaid by information placed there during compilation of a subsequent program. Then, when the object modules are link edited and executed, the information recorded during compilation will not be available if it has been overlaid.

To avoid the loss of the information needed to produce a symbolic formatted dump, assign a unique symbolic debugging work file for each COBOL source program to be compiled. On a mass storage device, these symbolic debugging work files must be assigned to non-overlapping extents. The following example shows how to use ASSGN statements to define the symbolic debugging work file on disk. Note that the compilation step assigns this data set to SYS005; at execution, it is assigned to SYS006.

```
// JOB SAMPLE
// OPTION SYM,NODECK,NOLISTX,NOXREF,LINK
// ASSGN SYS005,X'192'
// DLBL DEMOXX,99/101,SD
// EXTENT SYS005,111111,1,0,1100,50
// EXEC FCOBOL
      CBL SYMDMP

   (COBOL source deck)

/*
// EXEC LNKEDT
// ASSGN SYS006,X'192'
// DLBL DEMOXX,99/101,SD
// EXTENT SYS006,111111,1,0,1100,50
// EXEC

   (SYMDMP control cards)

/*
/&
```

If the symbolic debugging work file is defined as a tape, one physical reel is required. The file-protect ring must be in the tape reel during compilation and execution.

FIPS Considerations

SYS006 is required only for FIPS processing. After printing the header and any CBL card options on SYSLST, the compiler writes the compilation listing on SYS006 as input to the FIPS phases. The FIPS output is written on SYSLST.

MACHINE CONFIGURATION FOR CMS

The minimum CMS virtual machine configuration will support the DOS/VS COBOL compiler and library.

1. A CMS virtual machine with a minimum of 320K bytes of virtual storage and with the CMS/DOS environment active is required. The compiler uses at least 60K bytes of CMS user storage.

2. System and Logical Units.

   SYSIN/SYSIPT must be assigned to the device that contains the input source file. SYSIN/SYSIPT can be assigned to a reader, tape, or disk.

   The user can issue the ASSGN command for the following units:

   SYSPCH to tape, punch, disk, or IGN
   SYSLST to tape, printer, disk, or IGN
   SYSLOG to terminal
   SYS001-SYS002 to disk
   SYS003-SYS005 to tape or disk
   SYS006 to disk

Note: CMS allows "disk" to also be the 3350 in native mode. CMS does not support the 3540 diskette, 2311 disks, or the 2321 datacell. If SYSIN/SYSIPT is unassigned at compilation time, the CMS/DOS COBOL interface issues an error message and terminates the FCOBOL command.

If SYSPCH, SYSLST, SYSLOG, or SYS0001-SYSnnn is unassigned at compilation time, the CMS/DOS COBOL interface sends the output to the SYSIN disk (if SYSIN is assigned to a read/write disk) or to the user's A-disk.

Note: SYSLNK must not be assigned; it is not supported as a system logical unit under CMS/DOS.

3. Device Support

   The following devices, which are supported by DOS/VS, are not supported by CMS/DOS:

   - Card Readers: 1442, 2560P, 2560S, 2596, 3504, 5425P, and 5425S.
   - Disks: 3540, 2311, 2321.
   - Printers: 2560P, 2560S, 3203, 3525, 5203, 5425P, and 5425S.
   - Other Devices: 3881 OMR, 3886OCR

Not all of the DOS/VSE LIOCS modules are supported in CMS/DOS; see the "LIOCS Modules" section of this manual for more information.

Execution-Time Considerations

If symbolic debugging is requested, an additional work file, SYS005, should be assigned at compile time. This file should also be assigned at execution time if symbolic debugging control statements are present, though not necessarily to SYS005. If the file is not assigned, the CMS/DOS COBOL interface directs the output to the SYSIN disk (if SYSIN is assigned to a read/write disk) or to the user's A-disk.

The SORT verb is not supported in CMS/DOS. Also, COBOL programs that use ISAM or segmentation cannot be executed under CMS/DOS.

WORK FILE REQUIREMENTS

This section describes the work files required by the compiler. In addition to SYSRES and SYSLNK, which must be assigned to disk units, the compiler may require up to six work files: SYS001, SYS002, SYS003, SYS004, SYS005, and SYS006. SYS005 is required only when the Symbolic Debug feature is requested. SYS006 is required for FIPS. Note: SYSLNK is not supported and must not be assigned for CMS/DOS.

SYS001 and SYS006 must be assigned to a disk unit. Each of the other data sets (SYS002, SYS003, SYS004, and SYS005) can be assigned to either a tape unit or a disk unit.

If the work files are on tape, each must be assigned to a separate tape unit. If the work files are on disk, the amount of disk work space required depends mainly on the size of the source program. The approximate percentages of tracks that should be assigned to each work file are shown in Figures 1 through 6. Estimates are for programs of approximately 1000 and 2100 source records. The programs are assumed to request a cross-reference listing, the symbolic debug feature, and the source program library facility (COPY and/or BASIS). Source programs with the library facility require the LIB option on the COBOL option card (CBL card). Programs using NOLIB require considerably less SYS004 work space.

For DOS/VSE, work file assignments for the compiler must be made in each partition in which the compiler operates. For example, if the compiler can operate in either the background or the foreground 1 (F1) partition, work file assignments must be made in both the background and F1. The CMS/DOS environment simulates only the background partition. CMS/DOS does not support the 2311.

Number of Source Records	Total Tracks	SYS001 %	SYS002 %	SYS003 %	SYS004 %	SYS005 %
1000	92	22	24	14	34	6
2100	146	16	19	14	44	7

Figure 1. Approximate Allocation of Work File Space -- IBM 2311

Number of Source Records	Total Tracks	SYS001 %	SYS002 %	SYS003 %	SYS004 %	SYS005 %
1000	44	21	18	14	38	9
2100	74	16	16	12	48	8

Figure 2. Approximate Allocation of Work File Space -- IBM 2314

Number of Source Records	Total Tracks	SYS001 %	SYS002 %	SYS003 %	SYS004 %	SYS005 %
1000	28	18	22	14	39	7
2100	46	15	17	13	48	7

Figure 3. Approximate Allocation of Work File Space -- IBM 3330

Number of Source Records	Total Tracks	SYS001 %	SYS002 %	SYS003 %	SYS004 %	SYS005 %
1000	43	19	19	14	39	9
2100	72	15	17	13	48	7

Figure 4. Approximate Allocation of Work File Space -- IBM 3340

Number of Source Records	Total Tracks	SYS001 %	SYS002 %	SYS003 %	SYS004 %	SYS005 %
1000	20	19	19	15	40	7
2100	34	16	17	13	49	7

Figure 5. Approximate Allocation of Work File Space -- IBM 3350

Number of Source Records	See Note below	SYS001 %	SYS002 %	SYS003 %	SYS004 %	SYS005 %
1000	1020	18	18	15	41	8
2100	1700	16	16	13	48	7

Note: Total Fixed Block Device (512 Bytes)

Figure 6. Approximate Allocation of Work File Space -- Fixed Block Device (BUFSIZE equals 512)

FIPS Work Files

The number of tracks required for SYS006 (the FIPS input file) depends on the size of the source program and compiler options specified. As basic output, each record contains a line of the compilation listing and is 121 characters in length. However, additional space is required whenever each of the following options is specified: SYM, LISTX or CLIST, XREF or SXREF. SYS006 also contains any error messages issued by the compiler.

LIOCS MODULES

The following lists contain the names of the preassembled LIOCS modules used by the compiler. These modules are supplied by IBM as part of the relocatable library when the IBM DOS/VSE System is distributed. Be sure that the modules listed are included in the system to which the compiler is added.

Certain preassembled LIOCS modules are required when cataloging IBM components to the core image library. Do not delete these modules from the relocatable library until after all the IBM components have been cataloged to the core image library and all program products have been installed. These modules are listed in "Required IOCS Modules."

Notes:

1. Additional LIOCS modules (for ASCII processing) are distributed with the DOS/VSE subroutines and are added to the relocatable library with the subroutines. These LIOCS modules are listed under "Relocatable Library" in the "Storage Requirements" section.

2. CMS/DOS does not support all the DOS/VSE logical transients, nor all the operands of the transients it does support. The CMS/DOS logical transient support is described in the VM/SP User's Guide.

The procedure for generating the compiler for use with CMS/DOS is exactly the same as the procedure for generating it for use with DOS/VSE: the LIOCS modules required are the same. CMS/DOS, during its open routines, checks that the file being opened is supported by CMS/DOS.

Relocatable Module Naming Conventions

Each module has an 8-character name. The name consists of a 3-character prefix and a 5-character field corresponding to the option permitted in generation of the module. The following 3-character prefixes identify the preassembled modules shipped by IBM:

IJC I/O Card (CDMOD)

IJD I/O Printer (PRMOD)

IJF I/O Magnetic Tape (MTMOD)

IJG Sequential Direct Access (SDMOD)

IJH	Index Sequential Direct Access (ISMOD)[1]
IJI	Direct Access Method (DAMOD)[1]
IJJ	Device Independent Access Method (DIMOD)
IJN	3540 Access Method (DUMOD)[1]

---

[1]ISMOD, DAMOD, and DUMOD are not supported for the CMS/DOS environment.

CDMOD

CDMOD name = IJCabcde

a = F RECFORM=FIXUNB (always for INPUT and CMBND files)
  = U RECFORM=UNDEF
  = V RECFORM=VARUNB

b = A CTLCHR=ASA (not specified CMBND)
  = C CCNTROL=YES
  = Y CTLCHR=YES
  = Z neither CTLCHR nor CONTROL is specified

c = B RDONLY=YES and TYPEFLE=CMBND
  = C TYPEFLE=CMBND
  = H RDONLY=YES and TYPEFLE=INPUT
  = I TYPEFLE=INPUT
  = N RDONLY=YES and TYPEFLE=OUTPUT
  = O TYPEFLE=OUTPUT

d = B WORKA=YES and IOAREA2=YES
  = I IOAREA2=YES
  = W WORKA=YES
  = Z neither WORKA nor IOAREA2 is specified
  = Z WORKA is not specified (for CMBND files only)

e = 0 DEVICE=2540
  = 1 DEVICE=1442
  = 2 DEVICE=2520
  = 3 DEVICE=2501
  = 4 DEVICE=2540 and CRDERR is specified
  = 5 DEVICE=2520 and CRDERR is specified
  = 6 DEVICE=3505
  = 7 DEVICE=3525 and FUNC omitted or FUNC=R or P
  = 8 DEVICE=2560 and FUNC omitted or FUNC=R or P
  = 9 DEVICE=5425 and FUNC omitted or FUNC=R or P
  = A DEVICE=3525 and FUNC=RP
  = B DEVICE=3525 and FUNC=RW
  = C DEVICE=3525 and FUNC=PW
  = D DEVICE=3525 and FUNC=I
  = E DEVICE=3525 and FUNC=RPW
  = F DEVICE=2560 and FUNC=RP
  = G DEVICE=2560 and FUNC=RW
  = H DEVICE=2560 and FUNC=PW
  = I DEVICE=2560 and FUNC=I
  = J DEVICE=2560 and FUNC=RPW
  = K DEVICE=5425 and FUNC=RP
  = L DEVICE=5425 and FUNC=RW
  = M DEVICE=5425 and FUNC=PW
  = N DEVICE=5425 and FUNC=I
  = O DEVICE=5425 and FUNC=RPW

CDMOD Names

IJCFAOI1	IJCFZOI5	IJCUZOI5	IJCVZOI9
IJCFAOI4	IJCFZOI7	IJCUZOI7	
IJCFAOI5	IJCFZOI8	IJCUZOI8	
IJCFAOI7	IJCFZOI9		IJCVZOZ1
IJCFAOI8	IJCFZOID	IJCUZOI9	IJCVZOZ4
IJCFAOI9	IJCFZOZ1	IJCUZOZ1	IJCVZOZ5
IJCFAOZ1	IJCFZOZ4	IJCUZOZ4	IJCVZOZ7
IJCFAOZ4	IJCFZOZ5	IJCUZOZ5	IJCVZOZ8
	IJCFZOZ7	IJCUZOZ7	IJCVZOZ9
	IJFCZOZ8	IJCUZOZ8	IJCVZOZF
IJCFAOZ5	IJCFZOZ9	IJCUZOZ9	IJCVZOZH
IJCFAOZ7	IJCFZOZA	IJCUZOZF	IJCVZOZI
IJCFAOZ8	IJCFZOZC	IJCUZOZH	IJCVZOZJ
IJCFAOZ9	IJCFZOZD	IJCUZOZI	IJCVZOZK
IJCFAOZA	IJCFZOZE	IJCUZOZJ	IJCVZOZM
IJCFAOZC	IJCFZOZF	IJCUZOZK	IJCVZOZN
IJCFAOZE	IJCFZOZH	IJCUZOZM	IJCVZOZO
IJCFAOZF			
IJCFAOZH	IJCFZOZI	IJCUZOZN	
IJCFAOZI	IJCFZOZJ	IJCUZOZO	
IJCFAOZJ	IJCFZOZK		
IJCFAOZK	IJCFZOZM		
IJCFAOZM	IJCFZOZN	IJCVAOI1	
IJCFAOZN	IJCFZOZO	IJCVAOI4	
IJCFAOZO			
IJCFCIZ0		IJCVAOI5	
IJCFCIZ1		IJCVAOI8	
IJCFCIZ2	IJCUAOI4	IJCVAOI9	
IJCFCIZ6	IJCUAOI5	IJCVAOZ1	
IJCFYOI0	IJCUAOI8	IJCVAOZ4	
IJCFYOI7	IJCUAOI9	IJCVAOZ5	
		IJCVAOZ8	
		IJCVAOZ9	
IJCFZII0	IJCUAOZ1		
IJCFZII1	IJCUAOZ4		
IJCFZII2	IJCUAOZ5	IJCVAOZF	
IJCFZII3	IJCUAOZ8	IJCVAOZH	
IJCFZII6	IJCUAOZ9	IJCVAOZI	
IJCFZII7	IJCUAOZF	IJCVAOZJ	
IJCFZII8	IJCUAOZH	IJCVAOZK	
IJCFZII9	IJCUAOZI	IJCVAOZM	
IJCFZIZ0	IJCUAOZJ	IJCVAOZN	
	IJCUAOZK	IJCVAOZO	
	IJCUAOZM		
IJCFZIZ1	IJCUAOZN		
IJCFZIZ2	IJCUAOZO	IJCVZII8	
IJCFZIZ3		IJCVZII9	
IJCFZIZ6		IJCVZIZ8	
IJCFZIZ7	IJCUZII8	IJCVZIZ9	
IJCFZIZ8	IJCUZII9	IJCVZIZF	
IJCFZIZ9	IJCUZIZ8	IJCVZIZG	
IJCFZIZA	IJCUZIZ9	IJCVZIZJ	
IJCFZIZB	IJCUZIZF	IJCVZIZK	
IJCFZIZE	IJCUZIZG	IJCVZIZL	
IJCFZIZF	IJCUZIZJ	IJCVZIZO	
IJCFZIZG	IJCUZIZK		
IJCFZIZJ	IJCUZIZL		
IJCFZIZK	IJCUZIZO	IJCVZOI1	
IJCFZIZL		IJCVZOI4	
IJCFZIZO		IJCVZOI5	
IJCFZOI1	IJCUZOI1	IJCVZOI7	
IJCFZOI4	IJCUZOI4	IJCVZOI8	

DAMOD

DAMOD name = IJIabcde

a = B RECFORM=UNDEF (handles both UNDEF and FIXUNB)
  = F RECFORM=FIXUNB
  = S RECFORM=SPNUNB
  = V RECFORM=VARUNB

b = A AFTER=YES
  = Z AFTER is not specified

c = E IDLOC=YES and FEOVD=YES
  = I IDLOC=YES
  = R FEOVD=YES
  = Z neither is specified

d = H ERREXT=YES and RELTRK=YES
  = P ERREXT=YES
  = R RELTRK=YES
  = Z neither is specified

e = W HOLD=YES and RDONLY=YES
  = X HOLD=YES
  = Y RDONLY=YES
  = Z neither is specified

DAMOD Names

IJIBAIRZ	IJIFZZZZ
IJIBAIZZ	IJISAIRZ
IJIBAZRZ	IJISAIZZ
IJIBAZZZ	IJISAZRZ
IJIBZIRZ	IJISAZZZ
IJIBZIZZ	IJISZIRZ
IJIBZZRZ	IJISZIZZ
IJIBZZZZ	IJISZZRZ
IJIFAIRZ	IJISZZZZ
IJIFAIZZ	
IJIFAZRZ	
IJIFAZZZ	
IJIFZIRZ	
IJIFZIZZ	
IJIFZZRZ	

DRMOD

```
DRMOD name = IJMZabD0

a = S  SETDEV = YES
  = Z  SETDEV = NO

b = R  RDONLY = YES
  = Z  RDONLY = NO
```

DRMOD Names

```
IJMZSRD0
IJMZSZD0
IJMZZRD0
IJMZZZD0
```

DUMOD

```
DUMOD name = IJNDabcZ

a = I  input
  = O  output

b = C  ERROPT=name

c = Z  RDONLY not specified
```

DUMOD Names

```
IJNDICZZ
IJNDOCZZ
```

ISMOD

ISMOD name = IJHabcde

a = A RECFORM=BOTH and IOROUT=ADD or ADDRTR
  = B RECFORM=FIXBLK and IOROUT=ADD or ADDRTR
  = U RECFORM=FIXUNB and IOROUT=ADD or ADDRTR
  = Z RECFORM is not specified and IOROUT=LOAD or RETRVE

b = A IOROUT=ADDRTR
  = I IOROUT=ADD
  = L ICROUT=LOAD
  = R IOROUT=RETRVE

c = B TYPEFLE=RANSEQ
  = G IOAREA2=YES and TYPEFLE=SEQNTL or IOROUT=LOAD
  = R TYPEFLE=RANDOM
  = S TYPEFLE=SEQNTL
  = Z neither is specified and IOROUT=LOAD or ADD

d = C CORINDX=YES
  = Z CORINDX is not specified

e = F CORDATA=YES, ERREXT=YES, and RDONLY=YES
  = G CORDATA=YES and ERREXT=YES
  = O CORDATA=YES and RDONLY=YES
  = P CORDATA=YES
  = S ERPEXT=YES and RDONLY=YES
  = T ERREXT=YES
  = Y RDONLY=YES
  = Z nothing is specified

ISMOD Names

IJHAARCP	IJHUABZZ
IJHAARCZ	IJHUARCP
IJHAARZP	IJHUARCZ
IJHAARZZ	IJHUARZP
IJHBABCP	IJHUARZZ
IJHBABCZ	IJHZLGZZ
IJHBABZP	IJHZLZZZ
IJHBABZZ	IJHZRECZ
IJHBARCP	IJHZRBZZ
IJHBARCZ	
IJHBARZP	IJHZRGZZ
IJHBARZZ	IJHZRRZZ
IJHUABCP	IJHZRSZZ
IJHUABCZ	
IJHUABZP	

MTMOD

MTMOD name = IJFabcde

a = F RECFORM=FIXUNB or FIXBLK
  = S RECFORM=SPNUNB or SPNBLK
  = U RECFORM=UNDEF
  = V RECFORM=VARUNB or VARBLK

  = X ASCII RECFORM=FIXUNB or FIXBLK
  = N ASCII RECFORM=UNDEF
  = R ASCII RECFORM=VARUNB or VARBLK

b = B READ=BACK
  = Z READ=FORWARD, or READ is not specified

c = C CKPTREC=YES
  = Z CKPTREC is not specified

d = W WORKA=YES
  = Z WORKA is not specified

e = M ERREXT=YES and RDONLY=YES
  = N ERREXT=YES
  = Y RDONLY=YES
  = Z neither is specified

MTMOD Names (See "Relocatable Library" in the "Storage Estimates" section for additional MTMOD modules.)

IJFFBZZN
IJFFZCZZ
IJFFZZZZ
IJFSZZWN
IJFUZZZN

IJFUZZZZ
IJFVZZZN
IJFVZZZZ

Name list for workfile type modules (TYPEFLE=WORK):

MTMOD name = IJFabcde

a = W always

b = E ERROPT=YES
  = Z ERROPT is not specified

c = N NOTEPNT=YES
  = S NOTEPNT=POINTS
  = Z NOTEPNT is not specified

d = Z always

e = M ERREXT=YES and RDONLY=YES
  = N ERREXT=YES
  = Y RDONLY=YES
  = Z neither is specified

System I/O Modules
(See "Required IOCS Modules" before deleting modules with the IJF prefix.)

IJFWEZZZ
IJFWZNZZ
IJFWZZZZ

PRMOD

PRMOD name = IJDabcde

a = F RECFORM=FIXUNB
  = V RECFORM=VARUNB
  = U RECFORM=UNDEF

b = A CTLCHR=ASA
  = Y CTLCHR=YES
  = C CONTRCL=YES
  = S STLIST=YES
  = T DEVICE=3525 with 2-line printer
  = U DEVICE=2560
  = V DEVICE=5425
  = Z neither CTLCHR nor CONTROL nor STLIST is specified

c = P PRINTOV=YES, DEVICE not 3525
  = I PRINTOV=YES, DEVICE=3525, and FUNC Omitted or FUNC=W[T]
  = F PRINTOV=YES, DEVICE=3525, and FUNC=RW[T]
  = C PRINTOV=YES, DEVICE=3525, and FUNC=PW[T]
  = D PRINTOV=YES, DEVICE=3525, and FUNC=RPW[T]
  = E ERROPT=YES and PRINTOV=YES not specified
  = Z PRINTOV=YES not specified, and DEVICE not 3525
  = O PRINTOV=YES not specified, DEVICE=3525, and FUNC omitted
       or FUNC=W[T]
  = R PRINTOV=YES not specified, DEVICE=3525, and FUNC=RW[T]
  = S PRINTOV=YES not specified, DEVICE=3525, and FUNC=PW[T]
  = T PRINTOV=YES not specified, DEVICE=3525, and FUNC=RPW[T]
  = U DEVICE=2560 or 5425 and FUNC=W or omitted
  = V DEVICE=2560 or 5425 and FUNC=RW
  = W DEVICE=2560 or 5425 and FUNC=PW
  = X DEVICE=2560 or 5425 and FUNC=RPW

d = I ICAREA2=YES
  = Z IOAREA2 is not specified

e = V RDONLY=YES and WORKA=YES
  = W WORKA=YES
  = Y RDONLY=YES
  = Z neither is specified

PRMOD Names

IJDFAPIZ	IJDUZPIZ	
IJDFAPZZ	IJDUZPZZ	
IJDFYPIZ	IJDVAPIZ	
IJDFYPZZ	IJDVAPZZ	
IJDFZPIZ	IJDVYPIZ	
IJDFZPZZ	IJDVYPZZ	
IJDUAPIZ	IJDVZPIZ	
IJDUAPZZ	IJDVZPZZ	
IJDUYPIZ		
IJDUYPZZ		
IJDFUUZZ	IJDVUUZZ	IJDUUUZZ
IJDFUUIZ	IJDVUUIZ	IJDUUUIZ
IJDFUVZZ	IJDVUVZZ	IJDUUVZZ
IJDFUVIZ	IJDVUVIZ	IJDUUVIZ
IJDFUWZZ	IJDVUWZZ	IJDUUWZZ
IJDFUWIZ	IJDVUWIZ	IJDUUWIZ

IJDFUXZZ	IJDVUXZZ	IJDUUXZZ
IJDFUXIZ	IJDVUXIZ	IJDUUXIZ
IJDFAOIZ	IJDVAOIZ	IJDUAOIZ
IJDFARIZ	IJDVARIZ	IJDUARIZ
IJDFASIZ	IJDVASIZ	IJDUASIZ
IJDFATIZ	IJDVATIZ	IJDUATIZ
IJDFAOZZ	IJDVAOZZ	IJDUAOZZ
IJDFARZZ	IJDVARZZ	IJDUARZZ
IJDFASZZ	IJDVASZZ	IJDUASZZ
IJDFATZZ	IJDVATZZ	IJDUATZZ
IJDFZOIZ	IJDVZOIZ	IJDUZOIZ
IJDFZRIZ	IJDVZRIZ	IJDUZRIZ
IJDFZSIZ	IJDVZSIZ	IJDUZSIZ
IJDFZTIZ	IJDVZTIZ	IJDUZTIZ
IJDFZOZZ	IJDVZOZZ	IJDUZOZZ
IJDFZRZZ	IJDVZRZZ	IJDUZRZZ
IJDFZSZZ	IJDVZSZZ	IJDUZSZZ
IJDFZTZZ	IJDVZTZZ	IJDUZTZZ
IJDFAIIZ	IJDUAIIZ	IJDVAIIZ
IJDFAFIZ	IJDUAFIZ	IJDVAFIZ
IJDFACIZ	IJDUACIZ	IJDVACIZ
IJDFADIZ	IJDUADIZ	IJDVADIZ
IJDFZIIZ	IJDUZIIZ	IJDVZIIZ
IJDFZFIZ	IJDUZFIZ	IJDVZFIZ
IJDFZCIZ	IJDUZCIZ	IJDVZCIZ
IJDFZDIZ	IJDUZDIZ	IJDVZDIZ
IJDFYIIZ	IJDUYIIZ	IJDVYIIZ
IJDFYFIZ	IJDUYFIZ	IJDVYFIZ
IJDFYCIZ	IJDUYCIZ	IJDVYCIZ
IJDFYDIZ	IJDUYDIZ	IJDVYDIZ
IJDFVUZZ	IJDVVUZZ	IJDUVUZZ
IJDFVUIZ	IJDVVUIZ	IJDUVUIZ
IJDFVVZZ	IJDVVVZZ	IJDUVVZZ
IJDFVVIZ	IJDVVVIZ	IJDUVVIZ
IJDFVWZZ	IJDVVWZZ	IJDUVWZZ
IJDFVWIZ	IJDVVWIZ	IJDUVWIZ
IJDFVXZZ	IJDVVXZZ	IJDUVXZZ
IJDFVXIZ	IJDVVXIZ	IJDUVXIZ

SDMOD

SDMODxx name = IJGabcde

a = C RECFORM=FIXUNB or FIXBLK and HOLD=YES
  = F RECFORM=FIXUNB or FIXBLK and HOLD is not specified
  = P RECFORM=SPNUNB or SPNBLK and HOLD=YES
  = Q RECFORM=SPNUNB or SPNBLK and HOLD is not specified
  = R RECFORM=UNDEF and HOLD=YES
  = S RECFORM=VARUNB or VARBLK and HOLD=YES
  = U RECFORM=UNDEF and HOLD is not specified
  = V RECFORM=VARUNB or VARBLK and HOLD is not specified

b = I SDMODxI
  = O SDMODxO
  = U SDMODxU

c = C ERROPT=YES and ERREXT=YES
  = E ERROPT=YES
  = Z neither is specified

d = M TRUNCS=YES and FEOVD=YES
  = T TRUNCS=YES
  = W FEOVD=YES
  = Z neither is specified

e = B CONTROL=YES and RDONLY=YES
  = C CCNTROL=YES
  = Y RDONLY=YES
  = Z neither is specified

SDMOD Names

IJGFIEWZ
IJGFOEWZ
IJGFUEWZ
IJGQIEWZ
IJGQOEWZ

IJGQUEWZ
IJGUIEWZ
IJGUOEWZ
IJGUUEWZ
IJGVIEWZ

IJGVOEWZ
IJGVUEWZ

NAME LIST FOR WORKFILE TYPE MODULES (TYPEFLE=WORK)

SDMODxx name = IJGabcde

a = T SDMODW specifies HOLD=YES
  = W SDMODW does not specify HOLD=YES

b = C ERROPT=YES and ERREXT=YES
  = E ERROPT=YES
  = Z neither is specified

c = N NOTEPNT=YES
  = R NOTEPNT=POINTRW
  = Z NOTEPNT is not specified

d = C CCNTROL=YES
  = Z CONTROL is not specified

e = T RDONLY=YES and UPDATE=YES
  = U UPDATE=YES
  = Y RDONLY=YES
  = Z neither is specified

System I/O Modules
(See "Required IOCS Modules" before deleting modules with the IJG prefix.)

IJGWEZZU
IJGWEZZZ
IJGWZNZZ
IJGWZRZZ

Required IOCS Modules

The following preassembled IOCS modules are required when cataloging IBM components to the core image library. These modules are also required when installing program products.

Module Names

IJFWEZZZ	IJJCPD1N
IJFWZNZZ	IJJCPD2
IJFWZZZZ	IJJCPD3
IJGFIETZ	IJJCPV
IJGWEZZU	IJJCPV1
IJGWEZZZ	IJJCPV2
IJGWZNZZ	IJJCP0
IJGWZRZZ	IJJCP0N
IJJCPA1N	IJJCP1
IJJCPDV	IJJCP1N
IJJCPDV1	IJJCP2
IJJCPDV2	IJJCP3
IJJCPD0	
IJJCPD0N	
IJJCPD1	

DOS/VS COBOL INSTALLATION -- DOS/VSE

This section describes DOS/VS COBOL Compiler and Library installation under DOS/VSE. The steps that follow present a brief overview of the installation process.

LIBRARIES REQUIRED -- DOS/VSE

The three libraries needed are:

- Core image library
- Source statement library
- Relocatable library

INSTALLATION PROCESS -- DOS/VSE

To install DOS/VS COBOL using MSHP (Maintain System History Program), take the following steps:

1. Use the INSTALL PRODUCT to install the compiler and/or library.
2. Optionally, code the CBL statement and place it in C.CBLOPTINS to change the compiler default options.

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

VERIFYING SUCCESS -- DOS/VS COMPILER INSTALLATION

A sample DOS/VS COBOL program (TESTRUN) to verify the success of the compiler installation is available in the source statement library. The sample program will be placed in the source statement library during the installation process.

APPLICATION OF CORRECTIVE MAINTENANCE

Three link-edit books are supplied with the modules installed in the relocatable library:

- ILACBVS for the compiler phases
- ILBDSYML for the SYMDMP library phases
- ILBD$LNK for the transient routines

It is recommended that these be used when link-editing the appropriate phases.

Note that if user-written linkage editor control statements are used for linkage editing, ACTION CLEAR must be specified.

DOS/VS COBOL INSTALLATION -- VM/SP

This section describes DOS/VS COBOL Compiler and Library installation under VM/SP. DOS/VS COBOL runs in the CMS DOS environment.

INSTALLATION PROCESS UNDER VM/SP CMS

To install DOS/VS COBOL under CMS, take the following steps:

1. Mount the distribution tape.
2. IPL the DOS/VSE system.
3. Run the DOS/VSE MSHP program to restore the private libraries (just as normal installation under DOS/VSE).

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

System Generation Considerations

The compiler and library can be used on a system only if the following parameters are specified for the SUPVR and FOPT macro instructions during system generation:

SUPVR:	SYSTEM=DISK	
	ASCII=YES	(required only when the ASCII features of the compiler are used)
	MPS=BJF	(required only when the compiler is to be executed in the foreground)
FOPT:	AB=YES	(for use of STATE, FLOW, STXIT, SYMDMP, and CCUNT features)
	PCIL=YES	(required only when the compiler is to be executed in the foreground and RELLDR=YES has not been specified)
	GETVIS=YES	(required for use of 3886; forced by VSAM=YES, and required if COUNT option is specified)
	RELLDR=YES	(if relocating loader is to be used)
	SYSFIL=YES	(if system files are to be on disk)
VSAM:	VSAM=YES	(required for use of the VSAM feature)

System options selected at system generation enable you to tailor the DOS/VSE compiler to fit your installation's needs. The system options that can be used to control compiler processing specify whether:

- Control statements are to be written on SYSLST.
- A dump is to be written on SYSLST if an abnormal termination occurs. (You may not want this if the SYMDMP, STATE, or FLOW features are used.)
- The object module produced by the compiler is to be link edited.
- An object deck is to be punched.
- The COBOL source statements are to be written on SYSLST.
- A Procedure Division map, Data Division map, or cross-reference listing is to be written on SYSLST.
- Diagnostic messages for the source program are to be written on SYSLST.

Instructions for specifying these options are given in the publications DOS/VSE System Generation and DOS/VS Compiler and Library Programmer's Guide.

CHANGING THE INSTALLATION DEFAULTS

To change the compiler default values to suit your installation, a new member, C.CBLOPTNS, must be added to the source statement library. This module must contain CBL and LST option cards defining the desired defaults. These may be overridden at compilation time by supplying a CBL and/or LST statement in the compiler input stream. CMS/DOS does not support the LST statement.

CBL Statement -- COBOL Option Control Statement

Although most options for compilation are specified either at system generation time or in the OPTION control statement, the COBOL compiler provides an additional statement, the CBL statement, for the specification of compile-time options unique to COBOL.

The CBL statement must be placed between the EXEC FCOBOL statement and the first statement in the COBOL program. The CBL statement cannot be continued. However, if specification of options will continue past column 71, more than one CBL card image may be used.

The options shown in the following format may appear in any order. No embedded blanks may appear in the operand field. Underscoring indicates the default value. No comments should appear in the operand field.

For additional information on the CBL statement, see DOS/VS COBOL Compiler and Library Programmer's Guide.

```
CBL   [ADV|NOADV]          [,APOST|,QUOTE]          [,BUF=nnnnn]
      [,CATALR|,NOCATALR]  [,CLIST|,NOCLIST]        [,COUNT|,NOCOUNT]
      [,FLAGE|,FLAGW]      [,FLOW[=nn]]             [,LANGLVL(1)|,LANGLVL(2)]
      [,LIB|,NOLIB]        [,LVL=A|B|C|D|,NOLVL]    [,OPTIMIZE|,OPT|,NOOPTIMIZE|,NOOPT]
      [,PMAP=h]            [,SEQ|,NOSEQ]            [,SPACEn]
      [,STATE|,NOSTATE]    [,STXIT|,NOSTXIT]        [,SUPMAP|,NOSUPMAP]
      [,SXREF|,NOSXREF]    [,SYMDMP[=filename]]
      [,SYNTAX|,CSYNTAX|,NOSYNTAX]  [,TRUNC|,NOTRUNC]  [,VERB|,NOVERB]
      [,VERBREF|,NOVERBREF]  [,VERBSUM|,NOVERBSUM]  [,ZWB|,NOZWB]
```

STORAGE REQUIREMENTS

This section defines the partition size required by the DOS/VSE compiler alone and the storage required by the compiler together with the DOS/VSE subroutine library on the core image library. The storage needed for the compiler, subroutines, and required IOCS modules on the relocatable library is also given.

VIRTUAL STORAGE

A minimum 64K bytes of virtual storage is required. The compiler needs added virtual storage if a BUF parameter is specified on the CBL card. Enough extra storage must be allocated to compensate for six buffers of that size.

CORE IMAGE LIBRARY

Number of Phases		Number of Library Blocks For AF3* (Physical Records)
Compiler	26	700
Transient routines	4	4
Library Phases	14	32
*All devices supported by Advanced Function, Release 3		

Compiler Phases

FCOBOL
FCOBOL04 (Note 5)
FCOBOL05
FCOBOL06 (Note 1)
FCOBOL08
FCOBOL10
FCOBOL11
FCOBOL12
FCOBOL20
FCOBOL21

FCOBOL22
FCOBOL25 (Note 2)
FCOBOL30
FCOBOL35 (Note 6)
FCOBOL40
FCOBOL45 (Note 7)
FCOBOL50
FCOBOL51
FCOBOL60 (Note 3)
FCOBOL62
FCOBOL63
FCOBOL64
FCOBOL65 (Note 2)
FCOBOL61
FCOBOL70
FCOBOL80 (Note 4)

Note 1: Phases FCOBOL05, FCOBOL06, and FCOBOL08 are executed only when the lister feature is used.

Note 2: Phases FCOBOL25 and FCOBOL65 are executed only when symbclic debugging is used.

Note 3: If optimization is not requested, phase FCOBOL60 is called instead of FCOBOL62, FCOBOL63, and FCOBOL64.

Note 4: If FIPS processing is not requested, phase FCOBOL80 is nct called, and compilation terminates after phase FCOBOL70.

Note 5: Phase FCOBOL05 is executed only if LIB option is set.

Note 6: Phase FCOBOL35 is executed only if USE FOR DEBUGGING is used.

Note 7: Phase FCOBOL45 is executed only if UNSTRING is used.

Transient Routines

$$BCOBEM (Executed only when SYMDMP, FLOW, STATE, or COUNT is specified.)
$$BCOBER
$$BCOBR1
$$BFCMUL

Library Phases

Executed only when SYMDMP is specified.

ILBDMP01
ILBDMP02
ILBDMP04
ILBDMP10
ILBDMP11
ILBDMP12
ILBDMP13
ILBDMP14
ILBDMP20
ILBDMP21
ILBDMP22
ILBDMP23
ILBDMP24
ILBDMP25

RELOCATABLE LIBRARY

Number of Modules		Number of Library Blocks (Physical Records)
Compiler	41	Approximately 3000
Library	95	Approximately 460
ICCS	7	Approximately 32

Compiler Modules

ILACBVS
ILACBL00
ILACBL01
ILACBL04
ILACBL05
ILACBL06
ILACBL08
ILACBL10
ILACBL11

ILACBL12
ILACBL20
ILACBL21
ILACBL22
ILACBL25

ILACBL30
ILACBL35
ILACBL40
ILACBL45
ILACBL50
ILACBL51
ILACBL60

ILACBL62
ILACBL63
ILACBL64
ILACBL65
ILACBL61
ILACBL70

ILACBL80
ILACBL81
ILACBL82
ILACBL83
ILACBL84
ILACBL85
ILACBL86
ILACBL87
ILACBL88
ILACBL89
ILACBL8A
ILACBL8B
ILACBL8C
ILACBL8D

Subroutines

Library Modules	Approximate Number of Bytes of Virtual Storage	Blocks
ILBDABX0	495	4
ILBDACP0	1050	7
ILBDACS0	242	2
ILBDADR0	350	4
ILBDANE0	325	3
ILBDANF0	110	2
ILBDASY0	90	2
ILBDATB0	260	2
ILBDBID0	115	2
ILBDBIE0	120	2
ILBDBII0	465	4
ILBDBUG0	1080	8
ILBDCKP0	850	6
ILBDCLK0	60	2
ILBDCLS0	150	2
ILBDCMM0	1032	7
ILBDCRD0	150	2
ILBDCT10	480	3
ILBDCVB0	1136	8
ILBDDAE0	345	4
ILBDDBG0	2805	16
ILBDDCI0	180	5
ILBDDIO0	720	5
ILBDDSP0	1330	9
ILBDDSR0	335	4
ILBDDSS0	810	6
ILBDDTE0	436	4
ILBDDUM0	2	2
ILBDEFL0	525	6
ILBDETB0	260	4
ILBDFLW0	1260	9
ILBDFMT0	185	3
ILBDFPW0	810	4
ILBDGDO0	160	2
ILBDGPW0	90	2
ILBDIDA0	400	3
ILBDIDB0	120	2
ILBDIDR0	1665	8
ILBDIDT0	695	4
ILBDIFB0	300	3
ILBDIFD0	160	5
ILBDIML0	90	2
ILBDINS0	1932	11
ILBDINT0	290	4
ILBDISE0	480	4
ILBDISM0	370	4
ILBDITB0	260	12
ILBDIVL0	75	2
ILBDMFT0	150	3
ILBDMNS0	375	4
ILBDMCV0	70	3
ILBDMRG0	1150	8
ILBDMVE0	250	3
ILBDNSL0	620	5
ILBDOCR0	1925	10

Library Modules	Approximate Number of Bytes of Virtual Storage	Blocks
ILBDCSY0	135	3
ILBDRCR0	150	2
ILBDRDI0	410	4
ILBDRDS0	240	3
ILBDRFM0	135	2
ILBDSAE0	400	4
ILBDSCH0	745	4
ILBDSEM0	280	5
ILBDSET0	300	3
ILBDSIO0	2856	17
ILBDSMV0	60	2
ILBDSPA0	3824	20
ILBDSRT0	3720	19
ILBDSTR0	80	3
ILBDSSN0	300	3
ILBDSTG0	672	6
ILBDSTI0	600	4
ILBDSTN0	1740	8
ILBDTAB0	475	4
ILBDTC00	1170	7
ILBDTC20	500	3
ILBDTC30	4220	19
ILBDTEF0	600	6
ILBDTOD0	200	2
ILBDTRN0	260	2
ILBDUPS0	110	2
ILBDUSL0	365	3
ILBDUST0	2176	12
ILBDUTB0	260	12
ILBDVBL0	315	4
ILBDVCO0	515	3
ILBDVI00	4660	22
ILBDVMO0	470	4
ILBDVOC0	4280	18
ILBDVTR0	140	2
ILBDWTB0	260	3
ILBDXDI0	275	3
ILBDXMU0	135	2
ILBDXPR0	670	6
ILBDXTN0	290	3

IOCS MODULES

Needed when the ASCII Support Feature is used.

IJFXBZZN
IJFXBZZZ
IJFXZZZZ
IJFNZZZN
IJFNZZZZ
IJFRZZZN
IJFRZZZZ

APPENDIX. DEVICE INFORMATION

This appendix gives information regarding specific input/output devices that can be used with a DOS/VS COBOL program.

MINIMUM AND MAXIMUM BLOCK SIZE VALUES

The minimum and maximum block sizes that can be specified for specific devices are shown in Figure 7.

Device Type	Fixed and Undefined Records Block Size (Bytes)		Variable Records Block Size (Bytes)	
	Minimum	Maximum	Minimum	Maximum
Card Reader	1	80	9	80
Card Punch	1	81	9	89
Print Line Length (1403, 3800, etc.)				
120 characters	1	121	9	129
132 characters	1	133	9	141
144 characters	1	145	9	153
150 characters	1	151	9	159
Magnetic Tape	18	32760	18	32760
Direct Access				
2314	1	7294	9	7294
3330	1	13030	9	13030
3340	1	8368	9	8368
3350	1	19069	9	19069
3375	1	32760	9	32760

Notes:

1. For DOS/VSE Fixed Block Architecture devices, see the manuals describing the devices you are using.
2. For direct access devices with the track overflow feature, the maximum is 32760 for each device.

Figure 7. DOS/VS COBOL Devices--Minimum and Maximum Block Sizes

SC28-6479-3

IBM DOS/VS COBOL IRM Printed in U.S.A. SC28-6479-3

IBM DOS/VS COBOL Compiler and
Library Installation Reference Material
SC28-6479-3

Reader's Comment Form

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. Comments may be written in your own language; English is not required.

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.*

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

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.

SC28-6479-3

Reader's Comment Form

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

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

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

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

IBM

IBM DOS/VS COBOL IRM Printed in U.S.A. SC28-6479-3