

Program Product

IBM FORTRAN IV (H Extended) Compiler and FORTRAN Library (Mod II) for OS and VM/370 (CMS) Installation Reference Material

**Program Numbers: 5734-FO3
5734-LM3**

The FORTRAN IV (H Extended) compiler and the FORTRAN IV Library (Mod II) are released as independent components that require installation under the IBM Operating System or the Conversational Monitor System component of the IBM Virtual Machine Facility/370.

This publication describes the procedures for installing the FORTRAN IV (H Extended) compiler and the FORTRAN IV Library (Mod II). It is intended for use by system programmers or planners who supervise the generation and maintenance of an installation's operating system.

IBM

Third Edition (November 1974)

This is a major revision of, and obsoletes, the previous edition, SC28-6861-1. Changes in this edition are listed in the Summary of Amendments, Number 2, on the facing page.

This edition applies to Release 2.1 of the FORTRAN IV (H Extended) Compiler and FORTRAN IV Library (Mod II).

Changes are periodically made to the specifications herein; any such changes will be reported in subsequent revisions or Technical Newsletters.

Requests for copies of 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, Programming Publications, 1271 Avenue of the Americas, New York, N.Y. 10020.

Date of Publication: November, 1972

Form of Publication: Revision, SC28-6861-1

CMS Installation Information

New: Programming Feature

With this release, the FORTRAN IV (H Extended) Compiler and the Mod II Library can be installed under CMS. Storage estimates information for the compiler and library under CMS has been added to the "Storage Estimates" section. A detailed step-by-step description of CMS installation procedures has been included in the "Installation Procedures" section. A sample compiler output listing has been added to Appendix B.

Revised Distribution Tape Format

Specification Change

To accommodate CMS files, the format of the distribution tape for the compiler and library has been revised. The new format is described in the "Installation Procedures" sections.

Additional Compiler Default Options

New: Programming Feature

The ADSIZE and CMSIZE options have been added to the FORTRAN macro instruction. They control the default size of the compiler address constant table (NADCON) and the backward connection table (CMAJOR). Information has been added to the section "Coding the FORTRAN Macro Instruction".

Editorial changes having no technical significance are not noted here.

Specific changes to the text as of this publishing date are indicated by a vertical bar to the left of the text. These bars will be deleted at any subsequent republication of the page affected.

Date of Publication: November, 1974

Form of Publication: Revision, SC28-6861-2

FORTRAN Macro Name Changed to FORTRANX

New: Programming and Documentation

The name of the FORTRAN macro has been changed to FORTRANX so that installations using more than one of the H level FORTRAN compilers will not have more than one installation macro with the same name.

New and Changed FORTRANX Macro Instruction Options

New: Programming and Documentation

The new options COMDUMP, COMOPTS, COMOVLY, COMPSYS, INSTERR, NAME, PERMXL, SIZE, SORFLAG, SORTERM, and TRACE, have been added to the FORTRANX macro instruction. Changes have been made to the values that may be specified in the existing OBJID, ADSIZE, and CMSIZE options.

Revised Discussion of Installation Procedure Messages

New: Programming and Documentation

The text of messages produced by the new FORTRANX macro instruction options has been included, and the discussion of compiler error messages has been deleted.

Revised Number of OS Files on Compiler Distribution Tape

New: Programming and Documentation

The number of files has been increased from 6 to 7 to include alternate OS cataloged procedures. Changes have been made to the discussion of user-written procedures to install the compiler and library.

CMS Installation Example

Maintenance: Programming and Documentation

An example showing how to install the compiler from a terminal has been included. Additionally, a number of minor changes to the procedure for installing the compiler have been made.

SYS1 FORTLIB Secondary Storage Requirements

Maintenance: Documentation Only

The number of directory records and 3330 disk tracks required for the SYS1.FORTLIB library has been increased.

SYSLIN DD Statement Blocking Factor

Maintenance: Documentation Only

A brief new section has been added describing how to change the blocking factor for the SYSLIN DD statement.

Editorial changes having no technical significance are not noted here.

Specific changes to the text as of this publishing date are indicated by a vertical bar to the left of the text. These bars will be deleted at any subsequent republication of the page affected.

This publication is designed for system programmers and planners who supervise the generation and maintenance of an installation's operating system. It describes the installation procedures required for the FORTRAN IV (H Extended) compiler and the FORTRAN IV Library (Mod II).

There are five major sections in this publication as follows:

- Introduction
- Installation Procedures
- Storage Estimates
- System Planning
- Appendixes

The "Introduction" contains an overview of the installation procedures and general information about the FORTRAN IV (H Extended) compiler and the FORTRAN IV Library (Mod II).

The "Installation Procedures" section is divided into two parts. The first part describes how to install the compiler under OS and VM/370 and the second part describes how to install the library under OS and VM/370.

The "Storage Estimates" section contains dynamic and secondary storage requirements for the compiler and library.

The "System Programming" section is divided into three parts. The first part describes how to modify OS Release 19 for installing the compiler and library. The second part describes how to create and alter the option table for the Extended Error Handling facility. The third part describes how to concatenate data sets using the link library list under OS.

The "Appendixes" section is divided into three parts. Appendix A lists the modules comprising the FORTRAN IV (H Extended) compiler and the FORTRAN IV Library (Mod II). In addition, information for

obtaining the module lengths is included. Appendix B describes the sample FORTRAN programs that are punched during the installation procedure to verify the installation of the FORTRAN IV (H Extended) compiler and the FORTRAN IV Library (Mod II). Appendix C lists diagnostic messages produced to check the coding of the FORTRAN and FORTLIB macro instructions.

REFERENCE PUBLICATIONS

It is assumed that users of the OS are familiar with the information in these publications:

IBM System/360 Operating System
Messages and Codes
Order No. GC28-6631

OS/VS Message Library:
VS1 System Messages
Order No. GC38-1001

OS/VS Message Library:
VS1 System Codes
Order No. GC38-1003

OS/VS Message Library:
VS2 System Messages
Order No. GC38-1002

OS/VS Message Library:
VS2 System Codes
Order No. GC38-1008

IBM System/360 Operating System
Operator's Reference
Order No. GC28-6691

Operator's Library:
OS/VS1 Reference
Order No. GC38-0110

Operator's Library:
OS/VS2 Reference
Order No. GC38-0210

IBM OS/MFT and OS/MVT Utilities
Order No. GC28-6586

IBM System/360 Operating System
Job Control Language Reference
Order No. GC28-6704

OS/VS JCL Reference
Order No. GC28-0618

OS/VS JCL Services
Order No. GC28-0617

IBM OS System Generation
Order No. GC28-6554

OS/VS1 System Generation Reference
Order No. GC26-3791

OS/VS2 System Generation Reference
Order No. GC26-3792

It is assumed that users of IBM VM/370 are familiar with the information in these publications:

IBM Virtual Machine Facility/370
Introduction
Order No. GC20-1800

IBM Virtual Machine Facility/370
Planning and System Generation Guide
Order No. GC20-1801

IBM Virtual Machine Facility/370 Command
Language User's Guide
Order No. GC20-1804

IBM Virtual Machine Facility/370
EDIT Guide
Order No. GC20-1805

IBM Virtual Machine Facility/370
Operator's Guide
Order No. GC20-1806

IBM Virtual Machine Facility/370
Terminal User's Guide
Order No. GC20-1810

Related Publications

The following publication contains general information about the FORTRAN IV (H Extended) compiler and Mod II library:

FORTRAN Program Products for OS and the
CMS Component of VM/370,
General Information
Order No. GC28-6884

Additional information pertaining to the use of the FORTRAN IV (H Extended) compiler and the FORTRAN IV Library (Mod II) once they are installed can be found in these publications:

IBM OS FORTRAN IV (H Extended)
Compiler Programmer's Guide
Order No. SC28-6852

IBM OS FORTRAN IV Library
Mathematical and Service Subprograms
Order No. GC28-6818

IBM OS FORTRAN IV Mathematical and
Service Subprograms
Supplement for Mod I and Mod II
Libraries
Order No. SC28-6864

IBM OS FORTRAN IV (H Extended)
Compiler and Library (Mod II)
Messages
Order No. SC28-6865

CONTENTS

INTRODUCTION	7
FORTTRAN IV (H Extended) Compiler	7
FORTTRAN IV Library (Mod II)	7
INSTALLATION PROCEDURES	9
FORTTRAN IV (H Extended) Compiler Under OS	9
Format of the FORTTRAN IV (H Extended) Compiler Distribution Tape	9
Requirements for Compiler Installation Under OS	9
Overview of the Compiler Installation Procedure Under OS	10
Installing the Compiler Under OS	12
Coding a User-written Procedure to Prepare the System for Loading the Distribution Tape	12
Coding the FORTTRANX Macro Instruction	14
Loading and Running the Distribution Tape Under OS	25
Running a Sample FORTTRAN Program to Verify Success of the Compiler Installation Under OS	26
FORTTRAN IV (H Extended) Compiler Under VM/370	27
Requirements for Compiler Installation Under CMS	27
Installing the Compiler Under CMS	27
FORTTRAN IV Library (Mod II) Under OS	37
Format of the FORTTRAN IV Library (Mod II) Distribution Tape	37
Requirements for Library Installation Under OS	37
Overview of the Library Installation Procedure Under OS	38
Installing the Library Under OS	40
Coding a User-Written Procedure to Prepare the System for Loading the Library Distribution Tape	40
Coding the FORTLIB Macro Instruction	42
Loading and Running the User-Written Procedure and the Library Distribution Tape Under OS	45
Running a Sample FORTTRAN Program To Verify the Success of the Library Installation and Updating SYS1.FORTLIB Under OS	46
FORTTRAN IV Library (Mod II) Under VM/370	47
Requirements for Library Installation Under CMS	47
For PTF Installation Only	48
Installing the Library Under CMS	48
Storage Requirements	56
SYSTEM PROGRAMMING	59
Compiler and Library Installation for OS Release 19 Users	59
Requirements for Installation on OS Release 19	59
Installation Procedures on OS Release 19	60
Copying the Extended Precision Floating Point Simulator Routines	60
Special Installation Considerations for System/360 Model 85 or 195, or System/370 Users with OS	62
Assembling User-Written Compiler and Library Modules on OS Release 19	63
Extended Error Handling Facility	64
How to Create or Alter an Option Table	64
The Link Library List Under OS	66
SYSLIN Definition in Cataloged Procedures	67
APPENDIX A: COMPILER AND LIBRARY MODULES	68
FORTTRAN IV (H Extended) Compiler Modules	68
FORTTRAN IV (H Extended) Compiler CMS Interface Module	68
Obtaining CSECT Sizes for the Compiler Modules Under OS	68
FORTTRAN IV Library (Mod II)	70
FORTTRAN IV Library (Mod II) CMS Interface Module	70
IHOCMS	70
APPENDIX B: SAMPLE PROGRAMS PRODUCED BY THE COMPILER AND LIBRARY DISTRIBUTION TAPES	72
FORTTRAN IV (H Extended) Compiler Under OS	72
FORTTRAN IV (H Extended) Compiler Under CMS	73
FORTTRAN IV Library (Mod II) Under OS	74

APPENDIX C: INSTALLATION PROCEDURE MESSAGES	77
FORTRAN IV (H Extended) Compiler	77
Installation Information Messages.	77
Installation Error Messages.	77
FORTRAN IV Library (Mod II)	80
Library Information Messages	80
Library Error Messages	80
INDEX	83

ILLUSTRATIONS

FIGURES

Figure 1. Files on the FORTRAN IV (H Extended) Compiler Distribution Tape	9
Figure 2. FORTRAN IV (H Extended) Compiler Installation Under OS	11
Figure 3. User-written Procedure for FORTRAN IV (H Extended) Compiler Installation .	12
Figure 4. Format of the FORTRANX Macro Instruction	16
Figure 5. Effects of Running the User-written Procedure and the Compiler Distribution Tape Under OS	26
Figure 5-1. Procedure for Installing the Compiler under CMS (Part 1 of 4)	33
Figure 6. Files on the FORTRAN IV Library (Mod II) Distribution Tape	37
Figure 7. FORTRAN IV Library (Mod II) Installation Under OS	39
Figure 8. User-written Procedure for Mod II Library Installation	40
Figure 9. Format of the FORTLIB Macro Instruction	43
Figure 10. Effect of Running the User-written Procedure and the Library Distribution Tape Under OS	46
Figure 11. Copying the Extended Precision Floating Point Simulator Routines from Release 20 into Release 19	61
Figure 12. Example of Assembler Language Macro Definition Used to Generate Option Table	65
Figure 13. Sample Output from the IMBMDMAP Service Aid Program	69
Figure 14. Sample Program Produced by the Compiler Distribution Tape Under OS	72
Figure 15. Sample Program Produced by the Compiler Distribution Tape Under CMS . . .	73
Figure 16. Sample Program Produced by the Library Distribution Tape (Part 1 of 2) . .	74

TABLES

Table 1. Minimum Dynamic Storage Requirements for Executing the FORTRAN IV (H Extended) Compiler Under OS and CMS	56
Table 2. FORTRAN IV (H Extended) Compiler Secondary Storage Requirements	57
Table 3. FORTRAN IV Library (Mod II) Secondary Storage Requirements	57
Table 4. FORTRAN IV (H Extended) Compiler and Library (Mod II) Secondary Storage Requirements under CMS	58
Table 4. Compiler Installation information messages (Part 1 of 2)	78
Table 5. Library Installation Information Messages	80

This publication contains installation procedures and reference material for system programmers and planners for installing the FORTRAN IV (H Extended) compiler and the FORTRAN IV Library (Mod II).

The FORTRAN IV (H Extended) compiler and the FORTRAN IV Library (Mod II) are problem programs that run under the IBM Operating System or the Conversational Monitor System (CMS) component of VM/370. They function independently of each other and are installed separately. Each program is distributed on its own tape that contains the compiler and library modules, control statements necessary for installation, and test programs for verifying the installation procedures. However, a user-written procedure, which is described in this publication, containing items that are dependent upon the facilities available at your installation is required to prepare the system prior to running the distribution tapes. Additional system preparation is required for the library when an option table must be created or altered for the Extended Error Handling feature.

FORTRAN IV (H EXTENDED) COMPILER

The FORTRAN IV (H Extended) compiler processes programs written in the FORTRAN language and produces object modules that are suitable as input to the linkage editor for subsequent execution on System/360 and System/370 machines. Optionally, the compiler is able to produce optimized object modules, perform precision conversion, process extended precision quantities, and process extensions to the FORTRAN IV language. Furthermore, a set of seven cataloged procedures provides a variety of compile, link edit, load, and execute control statements for the compiler. For information on using the compiler once it is installed see the publications OS FORTRAN IV (H Extended) Programmer's Guide, Order No. SC28-6852 or IBM Virtual Machine Facility/370 Terminal User's Guide for FORTRAN Program Products, Order No. SC28-6891.

FORTRAN IV LIBRARY (MOD II)

The FORTRAN IV Library (Mod II) contains mathematical, service, and interface routines. The library is designed to support the extended precision, automatic precision increase, and asynchronous input/output (for OS only), features of the FORTRAN IV language. For information describing the various Mod II library routines, see the publications OS FORTRAN IV Library -- Mathematical and Service Subprograms, Order No. GC28-6818 and OS FORTRAN IV Mathematical and Service Subprograms Supplement for Mod I and Mod II Libraries, Order No. SC28-6864.

FORTRAN IV (H EXTENDED) COMPILER UNDER OS

FORMAT OF THE FORTRAN IV (H EXTENDED) COMPILER DISTRIBUTION TAPE

Figure 1 describes the format of the distribution tape.

File Number	Contents
1	OS JCL
2	CMS File
3	FORTRANX Options Macro
4	Compiler Object (TEXT) Decks
5	OS Cataloged Procedures
6	Alternate OS Cataloged Procedures
7	OS FORTRAN Sample Job

Figure 1. Files on the FORTRAN IV (H Extended) Compiler Distribution Tape

REQUIREMENTS FOR COMPILER INSTALLATION UNDER OS

- A System/360 or System/370 machine that can support the OS/MFT, OS/MVT, or OS/VS environments.
- The distribution tape for the FORTRAN IV (H Extended) compiler.
- An installed Release 20 and subsequent releases of OS or a release of OS/VS. (OS Release 19 users see the section "Compiler and Library Installation for OS Release 19 Users" for specific requirements.)
- 176K bytes of main storage for processing the compiler distribution tape.
- Space available on SYS1.LINKLIB or a private library for the FORTRAN IV (H Extended) compiler (see Table 2 in the "Storage Estimates" section for SYS1.LINKLIB storage requirements).
- Space available on SYS1.PROCLIB or a private library for the FORTRAN IV (H Extended) cataloged procedures (see Table 2 in the "Storage Estimates" section for SYS1.PROCLIB storage requirements).
- The IEBUPDTE and IEBPROGM utility programs.

OVERVIEW OF THE COMPILER INSTALLATION PROCEDURE UNDER OS

Compiler installation consists of the following steps:

1. Coding a user-written procedure to prepare the system for loading the compiler distribution tape.
2. Coding the FORTRANX macro instruction
3. Loading and running the user-written procedure and the compiler distribution tape.
4. Running a sample FORTRAN program to verify the success of the compiler installation.

Figure 2 illustrates the overall installation procedure. The steps for installing the compiler are discussed in detail in the following paragraphs.

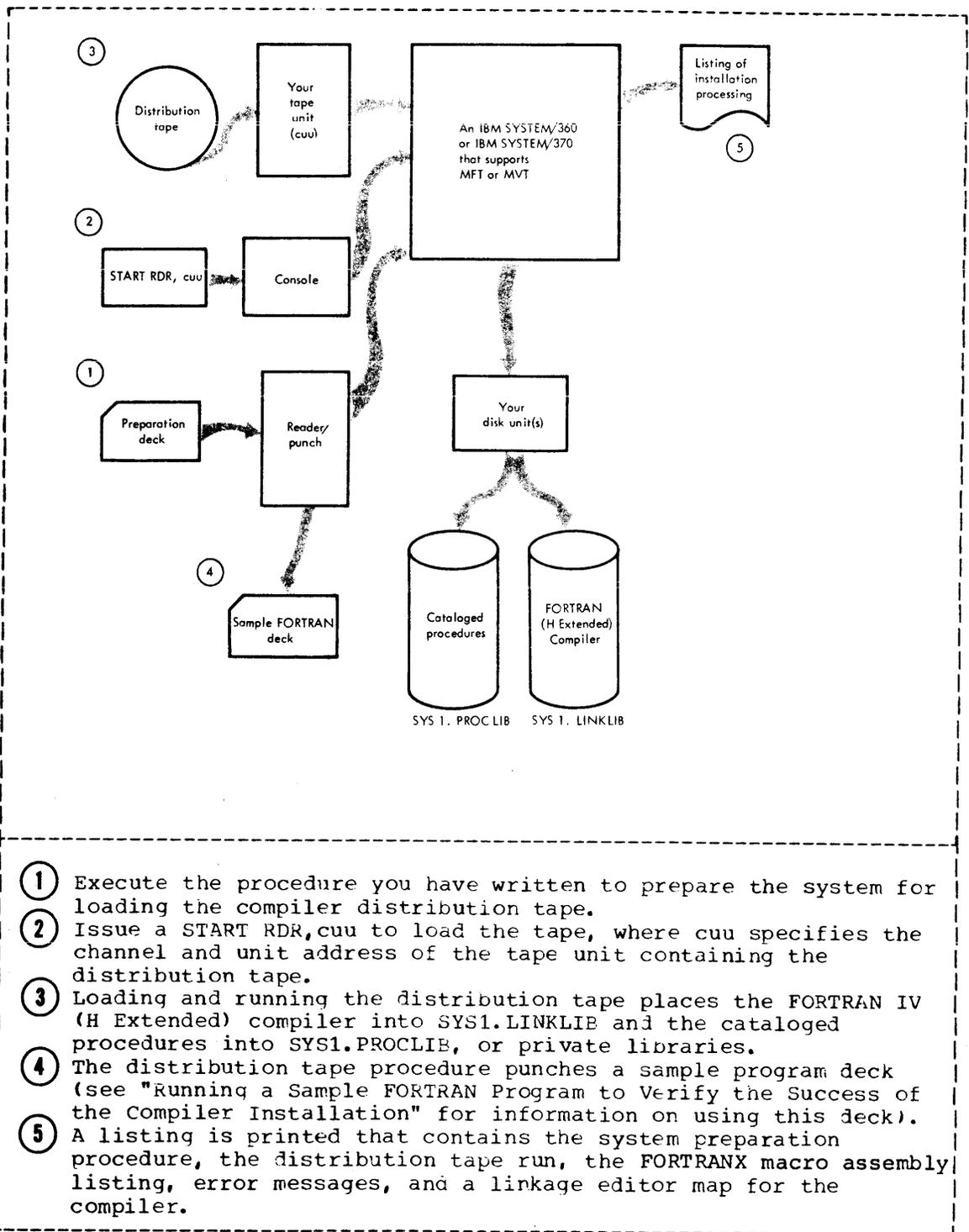


Figure 2. FORTRAN IV (H Extended) Compiler Installation Under OS

INSTALLING THE COMPILER UNDER OS

CODING A USER-WRITTEN PROCEDURE TO PREPARE THE SYSTEM FOR LOADING THE DISTRIBUTION TAPE

Before you can run the distribution tape, you must prepare and run a card deck containing a 2-step job. The first step places into SYS1. PROCLIB a cataloged JCL procedure that identifies the procedure and linkage libraries that will contain the FORTRAN IV (H Extended) cataloged procedures and compiler. The second step specifies the FORTRAN IV (H Extended) compiler default options that are to be installed using the FORTRANX macro instruction.

Figure 3 illustrates the 2-step job that is required to prepare the system for installing the compiler. The lettered statements contain fields (shown in lower case) that you must code based upon the facilities available at your installation.

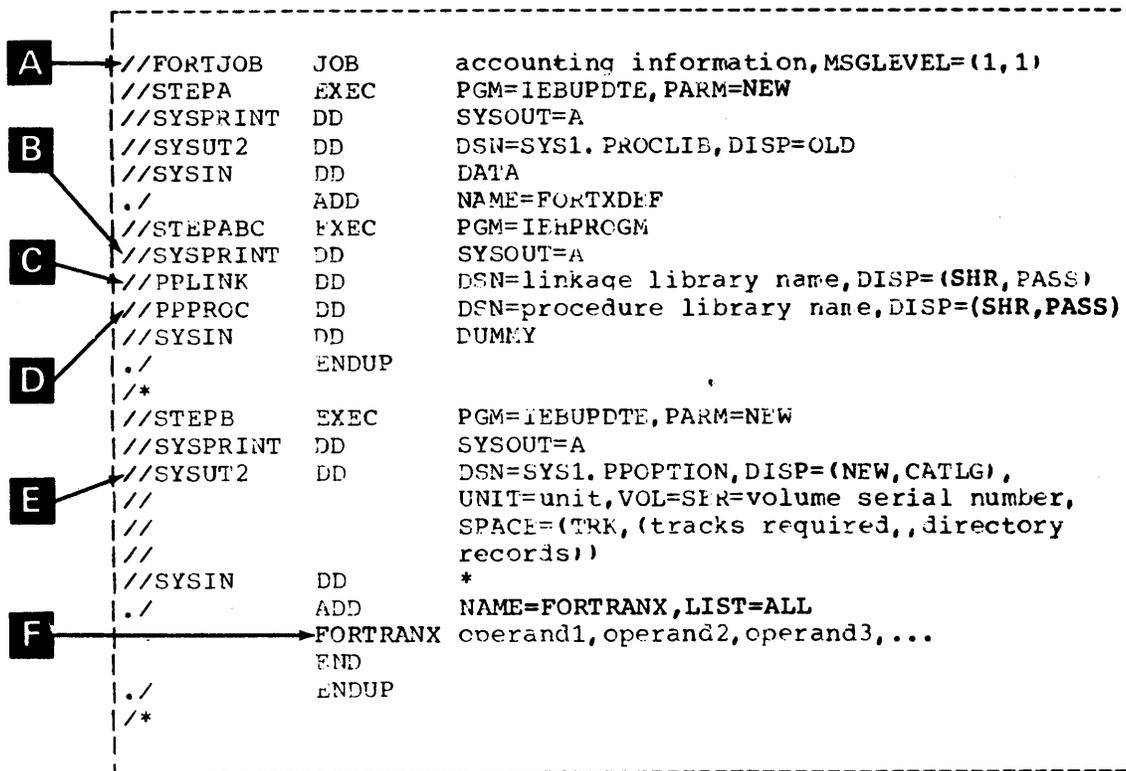


Figure 3. User-written Procedure for FORTRAN IV (H Extended) Compiler Installation

An explanation of the lettered statements in Figure 3 follows:

- A** Supply any accounting information that your installation requires.
- B** If you wish to block this output, add `DCB=BLKSIZE=value` to this statement.
- C** Insert the name of the linkage library on which the compiler will reside (either `SYS1.LINKLIB` or a private linkage library). If the library you choose is not cataloged, add to this statement the volume serial number on which it resides.

Note: The FORTRAN IV (H Extended) cataloged procedures assume that the compiler resides on `SYS1.LINKLIB`. If you are using a private library, you may concatenate it with `SYS1.LINKLIB` at initial program load (IPL) time or you may include in the JCL of any FORTRAN program that will use the compiler, a `JOBLIB` or `STEPLIB DD` statement that describes the private library on which the compiler resides. See the "Link Library List" section for information on concatenating data sets with `SYS1.LINKLIB` or the publication IBM System/360 Operating System Job Control Language Reference, Order No. GC28-6704, for information on using the `JOBLIB` and `STEPLIB DD` statements.

- D** Insert the name of the procedure library on which the cataloged procedures will reside (either `SYS1.PROCLIB` or a private procedure library).
- E** To allocate space for the `SYS1.PPOPTION` data set, indicate the direct-access unit and volume serial number on which it will reside. Using Table 2 in the "Storage Estimates" section, determine the number of tracks and directory records required by your device for `SYS1.PPOPTION`.

Note: If a `SYS1.PPOPTION` data set already exists in your system, the `STEPB SYSUT2 DD` statement must be replaced by:

```
//SYSUT2 DD DSN=SYS1.PPOPTION,DISP=OLD
```

- F** Code the `FORTRANX` macro instruction, specifying the compiler options that you wish to install as system default options (see the section "Coding the `FORTRANX` Macro Instruction" for more detailed information).

CODING THE FORTRANX MACRO INSTRUCTION

The FORTRANX macro instruction establishes the system defaults for the compiler options that can be specified by the FORTRAN programmer in the PARM parameter of the EXEC statement. These defaults will be assumed if the parameters are not coded in the PARM parameter by the FORTRAN programmer. In coding the FORTRANX macro instruction, all keyword operands (PUNCH, SORLIST, STORMAP, etc.) are optional and have their own defaults (the underlined values in Figure 4). For example, if the FORTRANX macro instruction is coded as follows:

FORTRANX STORMAP=MAP,LINECNT=55,OPT=1,OBJID=GOSTMT

the following system defaults for the FORTRAN IV (H Extended) compiler options are established:

<u>Coded Default</u>	<u>Generated Default</u>	<u>Explanation</u>
	NODECK	PUNCH operand was not coded.
	SOURCE	SORLIST operand was not coded.
MAP		STORMAP operand specified MAP.
	OBJECT	OBJPROG operand was not coded.
	EBCDIC	SORCODE operand was not coded.
LINECNT=55		LINECNT operand specified 55.
	ADSIZE=4096	ADSIZE operand was not coded.
	CMSIZE=4088	CMSIZE operand was not coded.
	NOLIST	OBJLIST operand was not coded.
GOSTMT		OBJIB operand specified GOSTMT.
OPT=1		OPT operand specified 1.
	NOFORMAT	SOREEDIT operand was not coded.
	NOXREF	SORXREF OPERAND was not coded.
	NOALC	SORALC operand was not coded.
	NOANSF	SORANSF operand was not coded.

<u>Coded</u> <u>Default</u>	<u>Generated</u> <u>Default</u>	<u>Explanation</u>
	COMPSYS=MVT	Compiler will not be executed in a virtual storage system; the operands MVT, TSO, VS1, VS2, or CMS were not coded.
	OVERLAY	NOOVERLAY operand was not coded.
	TERMINAL	NOTERMINAL operand was not coded.
	SORFLAG=I	All informational and diagnostic messages will be printed; the operands E or S were not coded.
	NOCOMPAT	COMPAT operand was not coded.
	NAME=MAIN	The name MAIN will be given to main programs, since no other name was coded.
	SIZE=MAX	The compiler will use all the storage space available to it, since no storage space was coded.
	NOLIST	LIST operand was not coded.
	NODUMP	DUMP operand was not coded.
	NOXL	XL operand was not coded.
	TRACE=0	No information concerning internal compiler tables will be printed, since no value was coded.

Furthermore, if all of the operand defaults are desired, code the FORTRANX macro without specifying any operands, as follows:

FORTRANX

In this case, all of the underlined values in Figure 4 will be established as the system defaults for this compiler options. The following paragraphs describe in detail the FORTRANX macro keyword operands and their respective parameters.

Name	Operation	Keyword Operands and Parameters
	FORTTRAN	PUNCH= { <u>DECK</u> } { <u>NODECK</u> } SORLIST= { <u>SOURCE</u> } { <u>NOSOURCE</u> } STORMAP= { <u>MAP</u> } { <u>NOMAP</u> } OBJPROG= { <u>OBJECT</u> } { <u>NOOBJECT</u> } SORCODE= { <u>BCD</u> } { <u>EBCDIC</u> } LINECNT= { <u>60</u> } { <u>number</u> } ADSIZE= { <u>size</u> } { <u>4096</u> } CMSIZE= { <u>size</u> } { <u>4088</u> } OBJLIST= { <u>LIST</u> } { <u>NOLIST</u> } OBJID= { <u>GOSTMT</u> } { <u>NOGOSTMT</u> } OPT= { <u>0</u> <u>1</u> <u>2</u> } SOREDIT= { <u>FORMAT</u> } { <u>NOFORMAT</u> } SORXREF= { <u>XREF</u> } { <u>NOXREF</u> } SORALC= { <u>ALC</u> } { <u>NOALC</u> } SORANSF= { <u>ANSF</u> } { <u>NOANSF</u> } COMPSYS= { <u>MVT</u> } { <u>MFT</u> } { <u>TSO</u> } { <u>VS1</u> } { <u>VS2</u> } { <u>CMS</u> } COMOVLY= { <u>OVERLAY</u> } { <u>NOOVERLAY</u> } SORTERM= { <u>TERMINAL</u> } { <u>NOTERMINAL</u> } SORFLAG= { <u>I</u> } { <u>E</u> } { <u>S</u> } COMOPTS= { <u>COMPAT</u> } { <u>NOCOMPAT</u> }

Note: All operands are optional; where a list of parameters is shown within braces, only one may be chosen. The underlined parameters are the defaults. Any number of operands may be coded. When coding, column 1 must be blank. The operation, FORTTRANX, may appear anywhere before column 72 but must precede the operands by at least one blank. The operands are separated by commas and may be continued on any number of cards as long as column 72 contains a nonblank character and the following card begins in column 16.

Figure 4. Format of the FORTTRANX Macro Instruction

Name	Operation	Keyword Operands and Parameters
	FORTRANX	NAME= {name} <u>MAIN</u> SIZE SIZE= <u>MAX</u> INSTERR= {LIST <u>NOLIST</u> } COMDUMP= {DUMP <u>NODUMP</u> } PERMXL= {XL <u>NOXL</u> } TRACE= {value} <u>0</u>

Note: All operands are optional; where a list of parameters is shown within braces, only one may be chosen. The underlined parameters are the defaults. Any number of operands may be coded. When coding, column 1 must be blank. The operation, FORTRANX, may appear anywhere before column 72 but must precede the operands by at least one blank. The operands are separated by commas and may be continued on any number of cards as long as column 72 contains a nonblank character and the following card begins in column 16.

Figure 4. Format of the FORTRANX Macro Instruction

PUNCH
controls 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.
If the PUNCH operand is not coded, NODECK is assumed.

SORLIST
controls the production of a listing of the FORTRAN source program.

SOURCE
specifies that the listing is to be produced.

NOSOURCE
specifies that the listing is not to be produced.
If the SORLIST operand is not coded, SOURCE is assumed.

STORMAP
controls the production of a map showing the relative location of variables, constants, etc., in the source program.

MAP
specifies that the map is to be produced.

NOMAP
specifies that the map is not to be produced.
If the STORMAP operand is not coded, NOMAP is assumed.

OBJPROG
the production of input to the linkage editor from the program being compiled.

OBJECT
specifies that the source program is to be processed by the linkage editor after compilation.

NOOBJECT
specifies that the source program is to be compiled only.
If the OBJPROG operand is not coded, OBJECT is assumed.

SORCODE
identifies the character set used to keypunch the source programs to be compiled.

BCD
specifies the BCD character set.

EBCDIC
specifies the EBCDIC character set.
If the SORCODE operand is not coded, EBCDIC is assumed.

LINECNT=number

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

number

is an integer from 01 to 200.

If the LINECNT operand is not coded, a value of 60 is assumed.

ADSIZE=size

specifies the length of the address constant table, NADCON, in compiler CSECT, IFEAAD. This is a fixed-size table whose size is determined when the compiler is installed. The size that is specified for the table affects the number of address constants, parameters, and temporaries that the compiler can handle during the compilation of a source program. Its maximum size is 16384 bytes and its minimum is 1024 bytes. The SIZE compiler option has no effect on the size of this table once installed. It only affects available work space.

size

is an integer from 1024 to 16384, or it may be specified as nK , where n is an integer from 1 to 16 and K represents 1024 bytes.

If the ADSIZE operand is not coded, or if ADSIZE=1, a value of 4096 is assumed.

CMSIZE=size

specifies the length of the backward connector table, CMAJOR, in compiler CSECT, IFEJAB. This is a fixed-size table whose size is determined when the compiler is installed. The size of the table affects the compiler's ability to perform certain optimization operations. The table receives backward connector information for each block in the source program (a block is a unit of instructions under one user or compiler generated label). If this table is too small to accommodate all the blocks in a source program, that compilation will be affected as follows:

- No branching optimization will be performed (affects OPTIMIZE(1) and (2) options).
- No text optimization will be performed (affects OPTIMIZE(2) option only).
- Register optimization will treat the entire program as a loop as it does for OPTIMIZE(1) (affects OPTIMIZE(2) option only).

The result is longer and less efficient object code. The SIZE compiler option has no effect on the size of this table once installed. It only affects available work space.

size

is an integer from 1024 to 65536, or it may be specified as nK , where n is an integer from 1 to 65 and K represents 1024 bytes.

If the CMSIZE operand is not coded or if CMSIZE=1, a value of 4088 is assumed.

OBJLIST
the production of a pseudo-assembler language listing of the object program.

LIST
specifies that the listing is to be produced.

NOLIST
specifies that the listing is not to be produced.

If the OBJLIST operand is not coded, NOLIST is assumed.

OBJID
controls the assignment of internal statement numbers to subroutine calls and function references.

GOSTMT
specifies that statement numbers are to be assigned.

NOGOSTMT
specifies that statement numbers are not to be assigned.

If the OBJID operand is not coded, NOGOSTMT is assumed.

OPT
controls the execution-time optimization of the object modules produced by the FORTRAN IV (H Extended) compiler.

0
specifies that the object module is not to be optimized.

1
specifies that the object module is to receive full register assignment and basic program optimization.

2
specifies that the object module is to receive full register assignment and complete program optimization.

If the OPT operand is not coded, 0 is assumed.

SOREDIT
controls the production of a structured source listing on the data set defined by the SYS\$PRINT DD statement. This listing indicates the loop structure and the logical continuity of the source program.

FORMAT
specifies that the listing is to be produced. OPT=2 must be specified with FORMAT.

NOFORMAT
specifies that the listing is not to be produced.

If the SOREDIT operand is not coded, NOFORMAT is assumed.

SORXREF

controls the production of a cross-reference listing on the data set defined by the SYSPRINT DD statement.

XREF

specifies that the listing is to be produced.

NOXREF

specifies that the listing is not to be produced.

If the SORXREF operand is not coded, NOXREF is assumed.

SORALC

determines whether or not variables in COMMON are to be aligned on fullword and doubleword boundaries.

ALC

specifies that boundary alignment of variables in COMMON is to be performed. For example, REAL*4 and LOGICAL*4 variables would be aligned on a fullword boundary, and COMPLEX*8 and REAL*8 variables would be aligned on a doubleword boundary. When boundary alignment is specified, gaps may appear in main storage.

NOALC

specifies that boundary alignment is not to be performed. If the SORALC operand is not coded, NOALC is assumed.

SORANSF

controls the processing of functions that are IBM extensions to the list American National Standard (ANS) FORTRAN functions.

ANSF

specifies that the compiler processes, as IBM-supplied, only those library and built-in functions that are on the list of ANS FORTRAN functions.

In this case, all non-ANS functions are treated as user-supplied and are processed accordingly. See the publication IBM System/360 and System/370 FORTRAN IV Language, Order No. GC28-6515 for the list of ANS functions.

NOANSF

specifies that the compiler processes, as IBM-supplied, all the library and built-in functions that are listed in the publication IBM System/360 and System/370 FORTRAN IV Language, Order No. GC28-6515. It should be noted that the ANS FORTRAN functions are a subset of these functions.

If the SORANSF operand is not coded, NOANSF is assumed.

COMPSYS

controls execution of the compiler and performs some optimization of compile space and time.

MVT, MFT, TSO

specify that the compiler will not be executed in a virtual storage system.

VS1, VS2

specify that the compiler will be executed in a virtual storage system.

CMS

specifies that the compiler will be executed under the control of CMS.

If the COMPSYS operand is not coded, MVT is assumed. However, under CMS, if the FORTRAN macro is not coded, CMS will be the assumed option. If the FORTRAN macro is coded, COMPSYS=CMS must be specified.

COMOVLY

controls the installation of the compiler and the choice of cataloged procedures under OS and VS.

OVERLAY

specifies that the compiler is to be link edited into an overlay structure.

NOOVERLAY

specifies that the compiler is to be link edited without an overlay structure.

If the COMOVLY operand is not coded, OVERLAY is assumed.

When OVERLAY is specified, the compiler region size in the cataloged procedures will be 256K. When NOOVERLAY is specified, the compiler region size in the cataloged procedure will be 600K.

(Note: The COMOVLY option does not apply to CMS.)

SORTERM

controls the production of a listing containing only compiler statistics and error messages to a terminal data set under CMS. Under OS and VS, this listing, which is produced on a data set whose ddname is SYSTEM, allows all messages for a FORTRAN batch compilation to appear together at the end of the compilation step (in addition to being interspersed between each two source listings).

TERMINAL

specifies that the listing is to be produced.

NOTERMINAL

specifies that the listing is not to be produced.

If the SORTERM operand is not coded, TERMINAL is assumed.

SORFLAG

controls the suppression of diagnostic messages.

I

specifies that informational diagnostic messages of level 0 and all diagnostic messages of higher level are to be printed. No diagnostic messages are to be suppressed.

E specifies that error diagnostic messages of level 8 and all diagnostic messages of higher level are to be printed. Informational messages (level 0) and warning diagnostic messages (level 4) are to be suppressed.

S specifies that severe error diagnostic messages of level 12 and unrecoverable error diagnostic messages of level 16 are to be printed. Informational messages (level 0), warning messages (level 4), and error diagnostic messages (level 8), are to be suppressed.

If the SORFLAG operand is not coded, I is assumed.

COMOPTS

controls the recognition of the following obsolete options by the compiler:

LOAD	NOLOAD	EDIT	NOEDIT	ID	NOID
OPT=	LINECNT=	SIZE=	NAME=		TRACE=

COMPAT

specifies that the obsolete options are to be recognized.

(Note: COMPAT may not be specified for the CMS system).

NOCOMPAT

specifies that the obsolete options are not to be recognized.

If the COMOPTS operand is not coded, NOCOMPAT is assumed.

(Note to OS and VS users: The COMPAT option allows both old and new formats of compiler options to be recognized. Installing with the NOCOMPAT option will mean that old job control (with the obsolete options) will have to be revised to the new format.

NAME=name specifies the name given to main programs.

name

is a string of one to six alphanumeric characters. This name will be used as the CSECT name for a main program only if it is the first module in the compilation.

If the NAME operand is not coded, a name of MAIN is assumed.

SIZE=size

specifies the amount of storage the compiler may use. It is used to confine the compiler and its work space to a portion of the space available and may be necessary if the compiler is invoked by another program. If MAX is specified, the compiler will make use of all the space available to it.

size

is specified as nK, where n is an integer from 150 to 99999 and K represents 1024 bytes, or the letters MAX.

If the SIZE operand is not coded, a value of MAX is assumed.

INSTERR

controls the listing of all possible installation error messages that can be produced by incorrect coding of the FORTRAN macro instruction keyword operands. If the LIST option is specified, the compiler will not be installed.

LIST

specifies that the listing is to be produced. The installation process will terminate after producing this listing without building a compiler.

NOLIST

specifies that the listing is not to be produced. The installation process is to proceed normally if no errors were encountered.

If the INSTERR operand is not coded, NOLIST is assumed.

The COMDUMP, PERMXL, and TRACE operands are intended for use only by IBM personnel responsible for program maintenance.

COMDUMP

controls the production of an ABEND dump when an unrecoverable error is encountered.

DUMP

specifies that a dump is to be produced.

NODUMP

specifies that a dump is not to be produced.

If the DUMP operand is not coded, NODUMP is assumed.

PERMXL

controls the ability to compile the FORTRAN modules which make up the FORTRAN H Extended compiler.

XL

specifies that the internal FORTRAN source modules can be compiled.

NOXL

specifies that the internal FORTRAN source modules cannot be compiled.

If the PERMXL operand is not coded, NOXL is assumed.

TRACE=value

controls the printing of internal compiler information and tables during the compilation process.

value

is an integer from 0 to 16777215. The meaning of the values is described in the publication IBM System/360 Operation System: Program Logic Manual for FORTRAN IV (H Extended), Order No. LY28-6403.

LOADING AND RUNNING THE DISTRIBUTION TAPE UNDER OS

The steps for running the distribution tape are as follows:

- ① Execute your user-written procedure to prepare the system for loading the tape. Running the procedure accomplishes the following:
 - Places the JCL procedure FORTXDEF into SYS1.PROCLIB. FORTXDEF identifies the libraries that will contain the compiler modules and the cataloged procedures (SYS1.LINKLIB and SYS1.PROCLIB or private libraries).
 - Places the FORTRANX macro instruction in SYS1.PPOPTION.
- ② Issue a START RDR,cuu to load the tape, where cuu specifies the channel and unit address of the tape unit containing the distribution tape. Running the distribution tape accomplishes the following:
 - Checks the coding of the FORTRANX macro instruction for valid syntax. See "Appendix C: Installation Messages" for diagnostic messages.
 - Assembles the compiler modules that depend on the macro instruction.
 - Link edits the FORTRAN IV (H Extended) compiler modules into SYS1.LINKLIB or a private linkage library.
 - Adds nine FORTRAN IV (H Extended) cataloged procedures to SYS1.PROCLIB, or a user-specified library.
 - Punches a card deck that contains a sample FORTRAN program that may be used to test the success of the compiler installation.

Figure 5 describes the effects of the two steps listed above. The numbers in the figure correspond to the numbered steps.

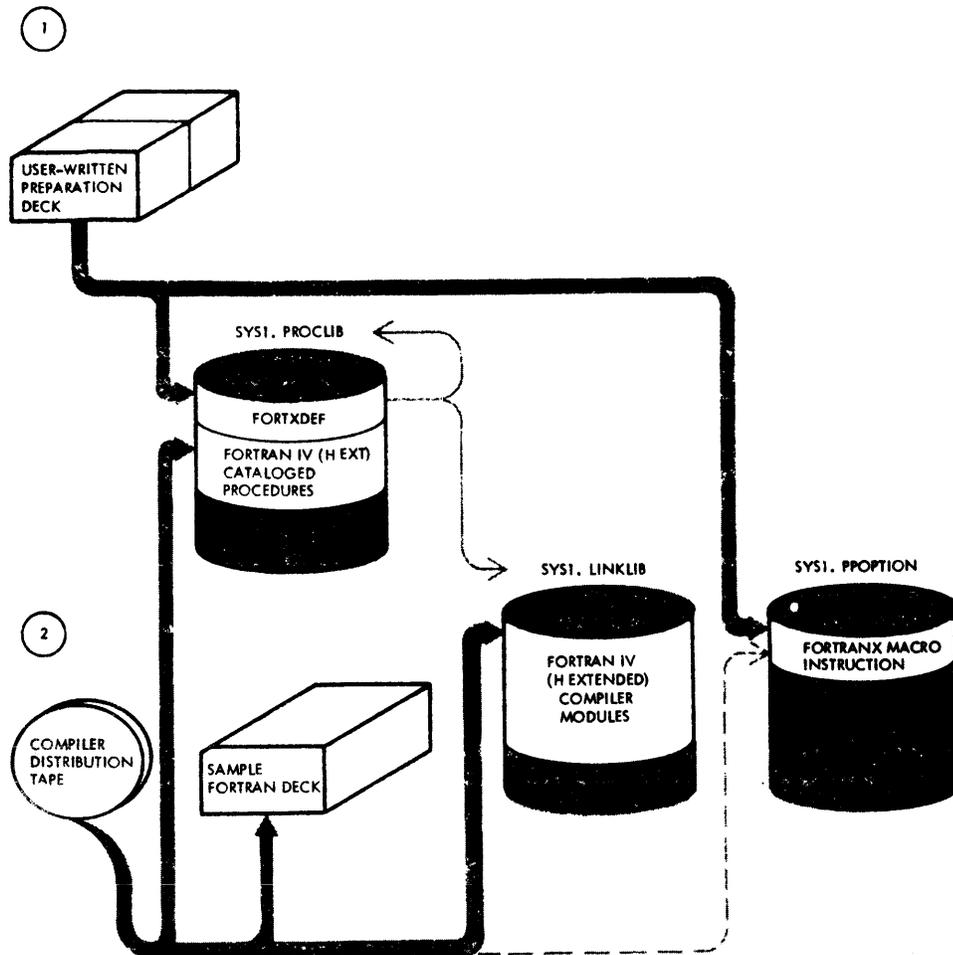


Figure 5. Effects of Running the User-written Procedure and the Compiler Distribution Tape Under OS

RUNNING A SAMPLE FORTRAN PROGRAM TO VERIFY SUCCESS OF THE COMPILER INSTALLATION UNDER OS

The distribution tape procedure punches a sample FORTRAN program that can be used to verify the compiler installation.

The procedure on the distribution tape also punches the JCL required to compile, link edit, and execute the sample FORTRAN program.

The cataloged procedure assumes that the FORTRAN IV Library (Mod II) resides on SYS1.FORTLIB. If it does not, you must include with the FORTRAN sample program a SYSLIB DD statement that identifies the data set on which the library resides.

FORTRAN IV (H EXTENDED) COMPILER UNDER VM/370

REQUIREMENTS FOR COMPILER INSTALLATION UNDER CMS

For Compiler and PTF Installation

- A System/370 machine that can support VM/370.
- An installed release of VM/370.
- A minimum CMS configuration with storage of 608K for the compiler only.
- The CMS library CMSLIB containing the extended arithmetic simulator routines IEAXPSIM IEAXPALL IEAXPDXR.
- The distribution tape for the FORTRAN IV (H Extended) compiler.

For Compiler Installation Only

- 542 blocks of unused space on your system (S) disk for the compiler (either a 2314 or the equivalent space on a 3330).
- Thirty (30) cylinders of unused space on a 2314 or the equivalent space on a 3330 for a scratch A disk.

For PTF Installation Only

- The following CMS files that were installed by the compiler distribution tape when the compiler was originally installed:

IFECMSFT	TXTLID
IFECMSFT	MACLIB
IFESTAL	EXEC
IFEOPT	EXEC
IFEGEND	EXEC
IFESAMP	FORTRAN
IFEHXGEN	ASSEMBLE
IFECMS	ASSEMBLE (optionally for rebuild)

Note: The above files must be on disks that have been made available through the ACCESS command.

INSTALLING THE COMPILER UNDER CMS

The compiler installation under CMS consists of the steps listed below. See Figure 5.1 at the end of this section for a graphic summary of steps 5 to 14 of this discussion.

- ① Mount the compiler distribution tape on virtual unit 181.
- ② Log into VM/370 and IPL CMS.

- ③ Issue an ACCESS command assigning the disk that you have selected as the scratch A disk.
- ④ Write and file the following EXEC procedure:

```

|edit prime exec
|EDIT;
|input
|INPUT;
|&control cms time
|tape rew
|&error &goto -errt
|tape fsf
|filedef inmove tap1 (recfm fb lrecl 80 block 80 change)
|filedef outmove disk cmsfort data (recfm f lrecl 80 block 80 change)
|&error &goto -err
|movefile
|cp spool punch to *
|punch cmsfort data (noheader
|read * *
|&begtype
|distribution tape is positioned beyond CMS file ready to issue
|ifestal
|&end
|cp spool punch off
|&exit
|-errt &type tape move error error exit
|&exit2
|-err &type error in move file exit
|&exit1
| CR
|EDIT:
|file

```

Note: It is advisable to keep this procedure permanently, in the event another installation is to be performed at a future date.

- ⑤ Type in PRIME to execute the PRIME procedure, which reads file 2 from the installation tape and makes the following CMS files available:

<u>Filename</u>	<u>Filetype</u>	<u>Contents</u>
IFESTAL	EXEC	installation control procedure
IFEOPT	EXEC	Compiler option macro instruction (IFEAAC ASSEMBLE) assembly procedure
IFEGEND	EXEC	Compiler build procedure
IFESAMP	FORTTRAN	Sample program for CMS
IFEHXGEN	ASSEMBLE	Source for the FORTRAN macro instruction
IFECMS	ASSEMBLE	Source for the compiler CMS interface routine
IFECMS	TEXT	Object code for the compiler CMS interface routine

- ⑥ Type in IFESTAL to execute the IFESTAL procedure, which will issue a QUERY command for your terminal and print the following message:

```
OPTION TO INSTALL OR REBUILD
DO YOU HAVE ALL NECESSARY FILES ON DISK?
IF RESPONSE 'YES' REBUILD
IF RESPONSE 'NO' READ INSTALLATION TAPE
```

- ⑦ If this is to be an installation, enter no. If you are installing PTFs, enter yes and skip to step ⑩.

Note: The alternative response of yes is for use when PTFs are to be applied to the compiler or system. When you receive a PTF you must remove the OS JCL either by punching out the PTF tape or by reading the tape into a CMS file and using the CMS editor. You must then insert the change into the appropriate TXTLIB for use by the installation procedure when rebuilding the compiler. For ease in segregating text decks for PTF installation, you will be requested in step ⑨ to name two additional macro libraries, which will be placed in front of IFECMSFT MACLIB with a GLOBAL command by the installation procedure. In addition, in step ⑫ you will be requested to name two additional text libraries, which will be placed in front of IFECMSFT TXTLIB by the installation procedure.

- ⑧ As a check, the IFESTAL procedure will type the following message at your terminal:

```
IS FORTRAN DISTRIBUTION TAPE MOUNTED ON VIRTUAL TAPE UNIT 181?
HAS THE CMS FILE BEEN READ OFF DISTRIBUTION TAPE?
IS DISTRIBUTION TAPE POSITIONED AFTER CMS FILE?
INSTALLATION OF 5734 FO3 FORTRAN IV H EXT
WHEN READY ENTER 'GO' OTHERWISE ENTER 'NOGO'
```

If the answer to the above questions is yes, enter go. If you have not done the items listed, enter noogo; do what is required, and re-execute the IFESTAL procedure.

- ⑨ During the installation of the compiler, the IFESTAL procedure types the following message at your terminal:

```
DO YOU WISH TO ASSEMBLE INTERFACE
RESPOND 'YES' OR 'NO' ON TERMINAL
```

If you want the interface assembled, enter yes; the interface routine will be assembled, an IFECMS TEXT file will be created, and a listing produced. If you answer no, the procedure checks to see that the compiler interface routine, IFECMS TEXT, is available. If it is not, the interface routine will be assembled; if it is available, no new IFECMS TEXT file or listing will be produced. During the installation of a PTF, this check is omitted and the message, shown above, does not appear.

Note: If you wish to force the assembly of the interface routine you must first erase an existing IFECMS TEXT file.

- ⑩ To provide a facility for segregating text decks (refer to step ⑦) during the installation of PTFs, the installation procedure will type the following message at your terminal:

```
-----  
ENTER CR 1 OR 2 MACLIB NAMES  
-----
```

If you do not want any additional MACLIBS, hit the RETURN key **CR**. If you want additional MACLIBS enter one or two names that will be used as the filename for the additional MACLIBS. They will be placed before the IFECMSFT MACLIB with a GLOBAL command issued by the installation procedure.

- ⑪ The IFESTAL procedure will continue processing and execute the IFEOPT procedure, which will type the following message at your terminal:

```
-----  
IF IFEHXGEN ASSEMBLE IS AS DESIRED FOR OPTIONS ENTER 'NOEDIT'  
IF YOU WISH TO EDIT THE OPTIONS STATEMENT THAT EXISTS ENTER  
'EDIT'  
TO EXIT THE CMS EDITOR YOU MUST ENTER CARRIAGE RETURN AND 'FILE'  
AFTER THE OPTIONS HAVE BEEN ENTERED  
TO EXIT INSTALL PROCEDURE IN CASE OF ERROR TYPE IN 'HALT'  
-----
```

See the section "Coding the FORTRANX Macro Instruction" for an explanation of the IBM-supplied default options. If all the defaults shown there are acceptable without any changes, type noedit and skip to step ⑬. If you want to change any of the IBM supplied defaults, type edit and go on to step ⑫.

- ⑫ The IFEOPT procedure will type the following list of IBM-supplied compiler option defaults:

```
-----  
* H EXT FORTRAN COMPILER OPTIONS DEFAULTS SHOWN  
| FORTRANX PUNCH=NODECK,SORLIST=SOURCE,STORMAP=NOMAP, *|  
| OBJPROG=OBJECT,SORCODE=EBCDIC,LINECNT=60, *|  
| OBJLIST=NOLIST,OPT=0,SOREEDIT=NOFORMAT, *|  
| SORXREF=NOXREF,OBJID=NOGOSTMT, *|  
| SORALC=NOALC,SORANSF=NOANSF, *|  
| NAME=MAIN,SORFLAG=I,SORTERM=TERMINAL, *|  
| COMPSYS=CMS,ADSIZE=4K,CMSIZE=4088,SIZE=MAX, *|  
| COMOPTS=NOCOMPAT,INSTERR=NOLIST, *|  
| COMDUMP=NODUMP,PERMXL=NOXL,TRACE=0 *|  
| *ENTER H EXT FORTRAN OPTIONS DESIRED *|  
-----
```

To change any of the defaults, type in the FORTRANX macro, following the conventions for editing a file with a filetype of ASSEMBLE.

- A** If the macro which you are about to type in contains continuation lines then do the following:

```
Hit the RETURN key  
Type: TRUNC 72  
Type: INPUT
```

(The system is now ready to accept your macro with continuation characters in column 72).

B Press the space bar to position the carrier at the beginning of the operation field. (At least one blank is required before the operation field.)

C Type in the macro name FORTRANX and each keyword operand to be changed together with its new default value.

Note: COMPSYS=CMS must be specified.

D If the number of options to be changed requires a continuation line, type in a character as a continuation indicator in column 72 (count from the left hand margin indicator), and continue typing the options beginning in column 16 of the following line.

E After all changes have been made, hit the RETURN key (**CR**) twice and issue the FILE subcommand.

Example:

```
-----  
|TAB fortranx stormap=map,sorxref=xref  
|CR  
|EDIT:  
|file  
-----
```

- ⑬ To provide a facility for segregating text decks (refer to step ⑦ for more information) during the installation of PTFs, the installation procedure will type the following message at your terminal:

```
-----  
|ENTER CR 1 OR 2 TXTLIB NAMES  
-----
```

If you do not want any additional TXTLIBS, hit the RETURN key, **CR**. If you want additional TXTLIBS enter one or two names that will be used as the filename for the additional TXTLIBS. They will be placed before the IFECMSFT TXTLIB with a GLOBAL command issued by the installation procedure.

- ⑭ The procedure will continue processing until the installation of the compiler and interface on the scratch disk is complete. It will compile the FORTRAN sample program, IFESAMP FORTRAN, and produce a listing, off-line, that contains the IFESAMP LISTING file that was produced for the sample program. You may verify the success of the compiler installation by requesting that this LISTING file be printed. The contents of this file are determined by the option you selected in step ⑫. The source program thus obtained should correspond to the program shown in Figure 15.

- ⑮ To move the compiler from the scratch system disk to your actual system disk, issue the following commands assuming that 191 is the scratch disk and 190 is the system disk.

```
-----  
|access 191 b  
|access 190 a  
|copy forthx module b2 = = a2  
|copy ieaxpsim module b2 = = a2  
|copy ieaxpall module b2 = = a2  
|copy ieaxpdxr module b2 = = a2  
|copy ifeaab module b2 = = a2  
|copy ifeatc module b2 = = a2  
|copy ifeatm module b2 = = a2  
-----
```

Note: You may save the contents of the scratch disk for your archives or to rebuild the compiler at a later date.

Figure 5.1 graphically summarizes steps 5 through 14 of the procedure described above.

```

* Install h ext compiler
* tape 181 has install tape mounted
* disk a has one file prime exec for reading in cms file
CMS
n disk a
A (391): 1 FILE; 10 PFC IN USE, 7966 LEFT (of 7976), 0% FULL (30 CYL), 3330, R/W
P;

tape rew
P;

prime
10:26:19 TAPE REV
10:26:21 TAPE FSF
10:26:22 FILEDEF MOVE TAP1 ( RECFM FB LRECL 80 BLOCK 80 CHANGE )
10:26:28 FILEDEF OUTMOVE DISK CMSFOPT DATA ( BLOCK 80 LRECL 80 RECFM F CHANGE )
10:26:35 MOVEFILE
**10:27:03 CP SPOOL PUNCH TO *
10:27:06 PUNCH CMSFOPT DATA ( NONHEADER
*PUN FILE 0180 TO M672JACP
10:27:13 READ * *
:READ IFEGEND EXEC A1 FORTPA 10/20/72 12:32.
:READ IFFSTAL EXEC A1 FORTPA 10/05/72 10:22.
:READ IFFOPT EXEC A1 FORTPA 10/05/72 10:15.
:READ IFECMS ASSEMBLE A1 FORTPA 10/24/72 16:21.
:READ IFFXGEN ASSEMBLE A1 FORTPA ?/16/72 16:46.
:READ IFFCMS TEXT A1 FORTPA 10/24/72 16:23.
:READ IFFSAMP FORTPAN A1 FORTPA 8/24/72 14:00.
DISTRIBUTION TAPE IS POSITIONED BEYOND CMS FILE READ TO ISSUE XXXSTAL
10:27:50 CP SPOOL PUNCH OFF
P;

spool printer cont
P;

* cms file has been read in ready to issue ifestall to install h ext compiler

```

Figure 5.1. Procedure for Installing the Compiler under CMS
(Part 1 of 4)

* Issue ifestal

```
ifestal
10:29:21 0 TIME
TIME IS 10:29:23 EST THURSDAY 05/30/74
CONNECT= 00:26:57 VIRTCPU= 000:14.82 TOTCPU= 001:08.47
10:29:24 0 TERMINAL
LINEFD # , LINEDEL & , CHARDEL @ , ESCAPE "
LINESIZE 129, MASK OFF, APL OFF, ATTN ON
OPTION TO INSTALL OR REBUILD
DO YOU HAVE ALL NECESSARY FILES ON DISK?
IF RESPONSE 'YES' REBUILD
IF RESPONSE 'NO' READ INSTALLATION TAPE
NO
IS FORTRAN DISTRIBUTION TAPE MOUNTED ON VIRTUAL TAPE UNIT 181?
HAS THE CMS FILE BEEN READ OFF DISTRIBUTION TAPE?
IS DISTRIBUTION TAPE POSITIONED AFTER CMS FILE?
INSTALLATION OF 5734 FORTRAN IV H EXT COMPILER
WHEN READY ENTER 'GO' OTHERWISE ENTER 'NOGO'
NO
10:30:20 FILEDEF INMOVE TAPI ( BLOCK 80 LRECL 80 RECFM FB )
10:30:26 FILEDEF OUTMOVE DISK IFFCNSET MACRO ( RECFM F BLOCK 80 LRECL 80 CHANGE )
10:30:32 MOVEFILE
*10:30:40 FILEDEF INMOVE TAPI ( RECFM FB LRECL 80 BLOCK 3200 CHANGE )
10:30:46 FILEDEF OUTMOVE DISK IFFCNSET TEXT ( RECFM F BLOCK 80 LRECL 80 CHANGE )
10:30:52 MOVEFILE
*****10:31:43 TAPE FSF
10:31:45 FILEDEF OUTMOVE DISK IFFCNSET FORTRAN ( RECFM F LRECL 80 BLOCK 80 CHANGE )
10:31:52 MOVEFILE
10:31:54 TAPE PFW

10:32:19 MACLIB GEN IFFCNSET IFFCNSET
10:32:22 TXTLIB GEN IFFCNSET IFFCNSET
*****BEGIN BUILD OF H EXT COMPILER
DO YOU WISH TO ASSEMBLE INTERFACE
RESPOND 'YES' OR 'NO' ON TERMINAL
NO
10:33:46 STATE IFFCNSET TEXT A1
10:33:48 EXEC IFFOPT
ENTER CP 1 OR 2 MACLIB NAMES

10:33:57 GLOBAL MACLIB IFFCNSET CMSLIB OSMACRO
IF IFFCNSET ASSEMBLE IS AS DESIRED FOR OPTIONS ENTER 'NOEDIT'
IF YOU WISH TO EDIT THE OPTIONS STATEMENT THAT EXISTS ENTER 'EDIT'
TO EXIT THE CMS EDITOR YOU MUST ENTER CARRIAGE RETURN AND 'FILE' AFTER
THE OPTIONS HAVE BEEN ENTERED
TO EXIT INSTALL PROCEDURE IN CASE OF ERROR TYPE IN 'HALT'
```

Figure 5.1. Procedure for Installing the Compiler under CMS
(Part 2 of 4)

```

edit
EDIT:

*      H EXT FORTRAN COMPILER OPTIONS DEFAULTS SHOWN
FORTRAN PUNCH=NODECK,SORLIST=SOURCE,STORMAP=NOIAP,
OBJPROC=OBJECT,SORCODE=EBDCIC,LINECNT=60,
OBJLIST=NOLIST,OPT=0,SOREPIT=NOFORMAT,
SORXREF=NOXREF,OBJID=NOGOSTMT
SORALC=NOALC,SORANSF=NOANSF,
NAME=MAIN,SORFLAG=1,SORTERM=TERMINAL,
COMPSYS=CMS,ADSIZE=4K,CMSIZE=4088,SIZE=MAX,
COMOPTS=NOCOMPAT,INSTERR=NOLIST,
COLDUMP=NODUMP,PERMXL=NOXL,TRACE=0
*
*      ENTER H EXT FORTRAN OPTIONS DESIRED

INPUT:
fortran  punch=nodeck,sorlist=nosource

EDIT:
file
$$$
ASSEMBLER (F) DONE

NO STATEMENTS FLAGGED IN THIS ASSEMBLY
10:37:35 EXEC IFFGEND
ENTER CP 1 OP 2 TXTLIB NAMES

10:37:41 GLOBAL TXTLIB IFFCNSF CMSLIB
10:37:44 LOAD IFAXPSIM ( CLEAR )
10:37:47 GENMOD IFAXPSIM MODULE A2 ( NOSTR )
10:37:51 PRINT LOAD MAP
10:37:53 LOAD IFAXPDYR ( CLEAR )
10:37:59 GENMOD IFAXPDYR MODULE A2 ( NOSTR )
10:38:03 PRINT LOAD MAP
10:38:05 LOAD IFAXPALL ( CLEAR )
10:38:08 GENMOD IFAXPALL MODULE A2 ( NOSTR )
10:38:12 INCLUDE IFFCMS ( CLEAR RESET IFFCMS )
10:38:16 GENMOD FORTHX MODULE A2 ( NOSTR MAP FROM IFFCMS )
10:38:21 INCLUDE IFFAAC ( CLEAR )
10:38:24 INCLUDE IFFAAR ( RESET IFFAAR )
10:39:08 GENMOD IFFAAR MODULE A2 ( STR NOMAP FROM IFFAAC )
10:39:20 RENAME LOAD MAP A5 LOADER MAP A5
10:39:23 LOAD IFAXPSIM
10:39:26 LOADMOD IFFAAR
10:39:36 ERASE LOAD MAP
10:39:38 RENAME LOADER MAP A5 LOAD MAP A5
10:39:42 INCLUDE IFFATH ( CLEAR RESET IFFATH )
10:39:46 GENMOD IFFATH MODULE A2 ( STR NOMAP FROM IFFATH )
10:39:51 INCLUDE IFFATC ( CLEAR RESET IFFATC )
10:39:55 GENMOD IFFATC MODULE A2 ( STR NOMAP FROM IFFATB )
10:40:00 PRINT LOAD MAP
10:40:02 ERASE LOAD MAP

```

Figure 5.1. Procedure for Installing the Compiler under CMS
(Part 3 of 4)

COMPILE SAMPLE PROGRAM
10:40:06 FORTHX IFFSAMP

FORTRAN II EXTENDED COMPILER ENTERED

STATISTICS SOURCE STATEMENTS = 10, PROGRAM SIZE = 390, SUBPROGRAM NAME = MAIN

STATISTICS NO DIAGNOSTICS GENERATED

***** END OF COMPILATION *****

373K BYTES OF CORE NOT USED

10:40:41 PRINT IFFSAMP LISTING
10:40:44 ERASE IFFSAMP LISTING
10:40:47 ERASE IFFSAMP TEXT
INSTALL OF II EXT COMPLETE SAMPLE JOB PRINTED OFFLINE
P;

* installation completed

1 * * a1 (date

FILENAME	FILETYPE	FM	FORMAT	PGS	BLOCKS	DATE	TIME
PRIME	EXEC	A1	F	80	30	4	5/30/74 10:24
CMSFORT	DATA	A1	F	80	1568	157	5/30/74 10:27
IFFGENC	EXEC	A1	F	80	40	4	5/30/74 10:27
IFFSTAL	EXEC	A1	F	80	147	15	5/30/74 10:27
IFFOPT	EXEC	A1	F	80	65	7	5/30/74 10:27
IFFCMS	ASSEMBLY	A1	F	80	1192	120	5/30/74 10:27
IFFHXGEN	ASSEMBLY	A1	F	80	8	1	5/30/74 10:27
IFFCMSSET	TXTLIB	A1	F	80	9015	902	5/30/74 10:37
IFFSAMP	FORTRAN	A1	F	80	11	2	5/30/74 10:27
IFFCMSSET	MACRO	A1	F	80	373	38	5/30/74 10:30
IFFCMSSET	TEXT	A1	F	80	8892	890	5/30/74 10:31
IFFCMSSET	FORTRAN	A1	F	80	55	6	5/30/74 10:31
IFFCMSSET	MACLIB	A1	F	80	374	38	5/30/74 10:37

P;

1 * module a102 (date

FILENAME	FILETYPE	FM	FORMAT	PGS	BLOCKS	DATE	TIME
IFAXPSIN	MODULE	A2	V	80	3	1	5/30/74 10:37
IFAXPPYR	MODULE	A2	V	1160	3	2	5/30/74 10:37
IFATC	MODULE	A2	V	3872	2	5	5/30/74 10:39
IFAXPALL	MODULE	A2	V	2760	3	4	5/30/74 10:38
FORTHX	MODULE	A2	V	5400	3	8	5/30/74 10:38
IFFAAR	MODULE	A2	V	65535	8	530	5/30/74 10:39
IFFATH	MODULE	A2	V	2576	2	4	5/30/74 10:39

P;

* move all files of type module to system disk to complete installation

spool printer nocont

P;

close printer

P;

* spooled printer continues at start of installation
* now get offline output printed

Figure 5.1. Procedure for Installing the Compiler under CMS
(Part 4 of 4)

FORTRAN IV LIBRARY (MOD II) UNDER OS

FORMAT OF THE FORTRAN IV LIBRARY (MOD II) DISTRIBUTION TAPE

Figure 6 describes the format of the distribution tape.

File No.	Contents	Record Characteristics		
		RECFM	LRECL	BLKSIZE
1	OS JCL	FB	80	80
2	CMS EXEC Procedure	FB	80	80
3	FORTLIB Options Macro	FB	80	80
4	Library Object (TEXT) Modules	FB	80	3200
5	Non-Extended Error Handling Object (TEXT) Modules	FB	80	3200
6	Extended Error Handling Object (TEXT) Modules	FB	80	3200
7	STAE	FB	80	3200
8	ADJST	FB	80	3200
9	OS FORTRAN Sample Job	FB	80	3200
10	EXCLUDE	FB	80	3200

Figure 6. Files on the FORTRAN IV Library (Mod II) Distribution Tape

REQUIREMENTS FOR LIBRARY INSTALLATION UNDER OS

You must have the following:

- A System/360 or System/370 machine that can support the OS/MFT (with the ATTACH option), OS/MVT or OS/VS environments.
- The distribution tape for the FORTRAN IV Library (Mod II).
- 176K bytes of main storage for processing the library distribution tape.
- An installed Release 20 or subsequent releases of OS or a release of OS/VS. (OS Release 19 users see the section "Compiler and Library Installation for OS Release 19 Users," for specific requirements.)
- Space available on SYS1.LINKLIB or a private linkage library for the IHOSTAE, IHOIOSUB, and IHOQERF2 modules.

Note: If boundary alignment is requested in the FORTLIB macro instruction, the IHOADJST module is included on SYS1.LINKLIB or a private library. See Table 3 in the "Storage Estimates" section for the storage required by SYS1.LINKLIB.

- Space available for the SYS1.FORTLIB data set or a private library, which will contain the remainder of the Mod II Library modules (see Table 3 in the "Storage Estimates" section for SYS1.FORTLIB storage requirements).
- 130 queue records for the system initiator/terminator.

Note: The number of queue records is determined at system generation time by the JOBQLMT parameter in the SCHEDULR macro instruction. If the value specified for JOBQLMT was less than 130, it must be raised to 130 during an initial program load (IPL) prior to installing the library. During the IPL, the operator must reply to the message:

```
|      id IEF423A SPECIFY JOB QUEUE PARAMETERS
```

with the following:

```
|      REPLY id,',130'
```

This reply is given in addition to any others that may be requested.

- The IEBUPDTE and IEHPROGM utility programs.

OVERVIEW OF THE LIBRARY INSTALLATION PROCEDURE UNDER OS

Library installation consists of the following steps:

1. Coding a user-written procedure to prepare the system for loading the library distribution tape.
2. Coding the FORTLIB macro instruction.
3. Loading and running the user-written procedure and the library distribution tape.
4. Running a sample FORTRAN program to verify the success of the library installation.

Figure 7 illustrates the overall installation procedure. The steps for installing the library are discussed in detail in the following paragraphs.

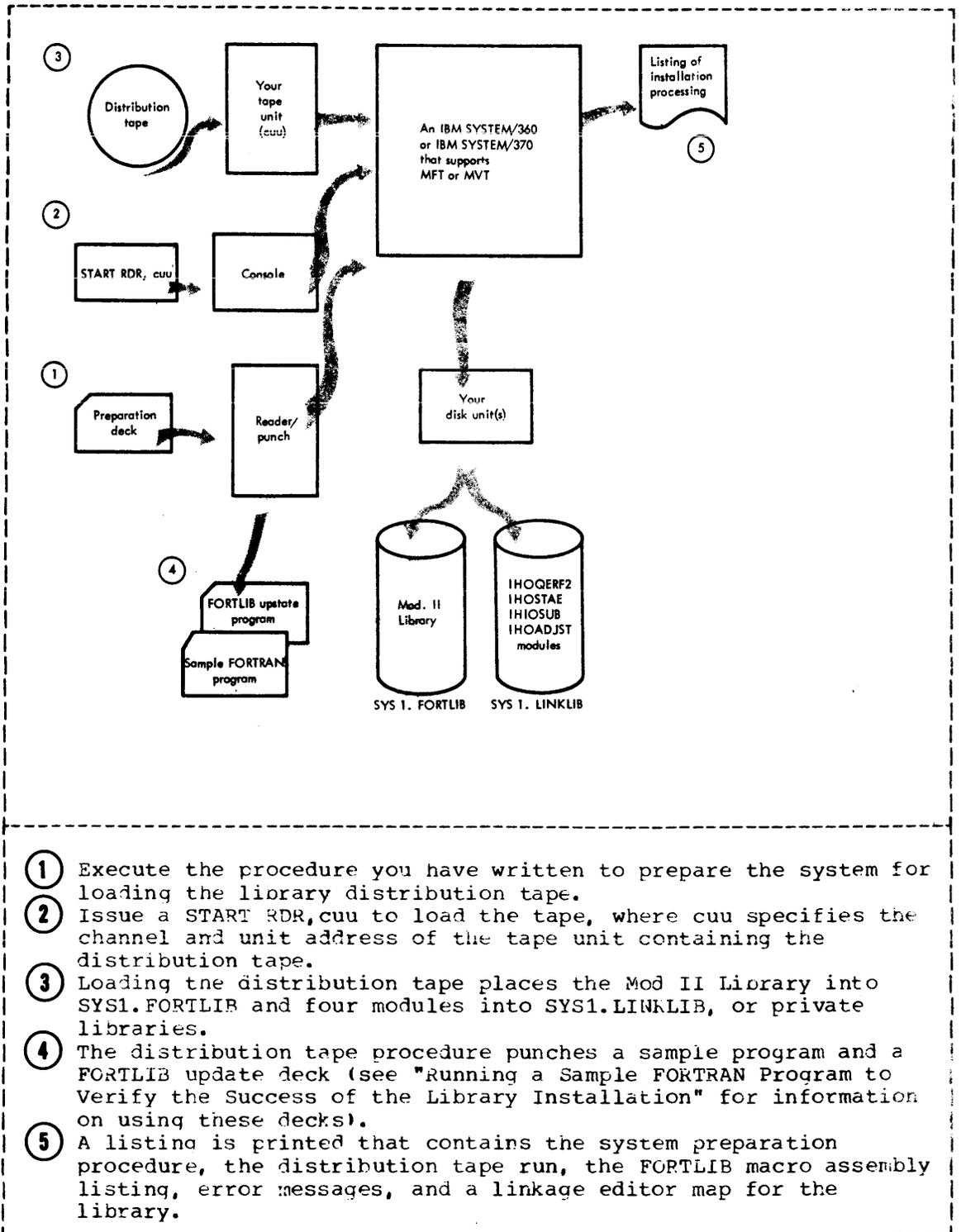


Figure 7. FORTRAN IV Library (Mod II) Installation Under OS

INSTALLING THE LIBRARY UNDER OS

CODING A USER-WRITTEN PROCEDURE TO PREPARE THE SYSTEM FOR LOADING THE LIBRARY DISTRIBUTION TAPE

Before you can run the library distribution tape, you must prepare a card deck containing a 2-step job. The first step places into SYS1.PROCLIB a cataloged JCL procedure that identifies the subprogram and linkage libraries that will contain the Mod II Library. The second step specifies the Mod II Library options that are to be installed using the FORTLIB macro instruction.

Figure 8 illustrates the 2-step job that is required to prepare the system for installing the library. The lettered statements contain fields (shown in lower case) that you must code based on the facilities available at your installation.

```

A  //LIBJOB   JOB      accounting information,MSGLEVEL=(1,1)
      //STEPA  EXEC     PGM=1EBUPDTE,PARM=MOD
      //SYSPRINT DD     SYSOUT=A
      //SYSUT1  DD     DSN=SYS1.PROCLIB,DISP=OLD
      //SYSUT2  DD     DSN=SYS1.PROCLIB,DISP=OLD
B  //SYSIN   DD     DATA
      .//      ADD     NAME=PPMODTWO
      //STEPABC EXEC    PGM=IEHPROGM
      //SYSPRINT DD     SYSOUT=A
C  //PPLINK  DD     DSN=linkage library name,DISP=(OLD,PASS)
      //PPFORT  DD     DSN=subprogram library name,
      //        DD     DISP=(OLD,PASS)
      //SYSIN   DD     DUMMY
      .//      DD     ENDUP
      /*
D  //STEPB   EXEC    PGM=1EBUPDTE,PARM=NEW
      //SYSPRINT DD     SYSOUT=A
      //DD1     DD     DSN=subprogram library name,
      //        DD     DISP=(NEW,PASS),
      //        DD     UNIT=unit,VOL=SER=volume serial number,
      //        DD     SPACE=(TRK,(tracks required,,directory
      //        DD     records))
E  //SYSUT2  DD     DSN=SYS1.PPOPTION,DISP=(NEW,CATLG)
      //        DD     UNIT=unit,VOL=SER=volume serial number,
      //        DD     SPACE=(TRK,(tracks required,,directory
      //        DD     records))
      //SYSIN   DD     *
      .//      ADD     NAME=FORTLIB,LIST=ALL
F  //        DD     FORTLIB operand 1,operand 2,operand 3,...
      //        DD     END
      .//      DD     ENDUP
      /*

```

Figure 8. User-written Procedure for Mod II Library Installation

An explanation of the lettered statements in Figure 8 follows:

- A** Supply any accounting information that your installation requires.
- B** Insert the name of the linkage library on which the library modules IHOADJUST, IHOICSUB, IHOQERF2, and IHOSTAE will reside (either SYS1.LINKLIB or a private library). If the library you choose is not cataloged, add to this statement the unit and volume serial number on which it resides.

Note: If you are using a private library, you may concatenate it with SYS1.LINKLIB at initial program load (IPL) time or you may include in the JCL of any FORTRAN program that will use the Mod II Library, a JOELIB or STEPLIB DD statement that describes the private library on which the Mod II Library resides. See the "Link Library List" section for information on concatenating data sets with SYS1.LINKLIB or the publication IBM System/360 Operating System Job Control Language Reference, Order No. GC28-6704, for information on using the JOELIB or STEPLIB statements.

- C** Insert the name of the subprogram library on which the Mod II Library will reside (either SYS1.FORTLIB or a private library). If you are using a private library, add to this statement the unit and volume serial number on which it resides. You may name your private library SYS1.FORTLIB; however, an existing SYS1.FORTLIB may not reside on the same volume as the new SYS1.FORTLIB. When you run a FORTRAN program that requires the Mod II Library, you must uncatalog the SYS1.FORTLIB you do not want to use.

Note: The FORTRAN IV (H Extended) cataloged procedures assume that the Mod II Library resides on SYS1.FORTLIB. If it does not, you must include in the JCL of any FORTRAN program that uses the Mod II Library, a JOELIB or STEPLIB DD statement that describes the private library on which the Mod II Library resides. See the publication IBM System/360 Operating System Job Control Language Reference, Order No. GC28-6704, for information on using the JOELIB or STEPLIB statements.

- D** To allocate space for your subprogram library, indicate the library name (same as in C, above), the direct-access unit, and volume serial number on which it will reside. Using Table 3 in the "Storage Estimates" section, determine the number of tracks required by your device for SYS1.FORTLIB.

Note: If an old SYS1.FORTLIB data set already exists in your system, before starting the distribution tape procedure, the data set must be uncataloged and located on a volume other than the one on which the new SYS1.FORTLIB will reside.

- E** To allocate space for the SYS1.PPOPTION data set, indicate the direct-access unit and volume serial number on which it will reside. Using Table 3 in the "Storage Estimates" section, determine the number of tracks and directory records required by your device for SYS1.PPOPTION.

Note: If a SYS1.PPOPTION data set already exists in your system, the STEPB SYSUT2 DD statement must be replaced by:

```
//SYSUT2 DD DSN=SYS1.PPOPTION,DISP=OLD
```

- F** Code the FORTLIB macro instruction, specifying the defaults you wish to install as system defaults for the Mod II Library. See the section "Coding the FORTLIB Macro Instruction" for more detailed information.

CODING THE FORTLIB MACRO INSTRUCTION

The FORTLIB macro instruction specifies input/output routine defaults, boundary alignment inclusion, and extended error handling option table information for the FORTRAN IV Library (Mod II).

The Mod II Library object-time input/output routines require information on the number of logical input/output units that are available to the system. The UNTABLE operand provides this information. These routines also require that defaults be established for the logical input/output units to be used for READ (on-line) statements, PUNCH (on-line) statements, error messages, and dumps. The ONLNRD, OBJERR, and ONLNPC operands establish the default data-set reference numbers to be used. The FORTRAN programmer using the library may use these defaults and does not need to supply a DD statement for these data sets. For example, if the FORTLIB macro instruction were coded as follows:

```
FORTLIB  UNTABLE=25,ONLNRD=15,OBJERR=16,ONLNPC=17
```

The following defaults would be established:

- Logical input/output units available - 25
- Logical input/output unit for READ statements - 15
- Logical input/output unit for error messages and dumps - 16
- Logical input/output unit for PUNCH statements - 17

A boundary alignment routine may be included in the Mod II Library. This routine corrects boundary alignment violations during object-time for variables in COMMON blocks and EQUIVALENCE groups and allows processing to continue. The BOUNDRY operand specifies whether the boundary alignment routine is to be included in the Mod II Library.

The extended error handling facility supplies the FORTRAN programmer with data-dependent or program errors that are detected during execution of his program. The OPTERR operand specifies whether the extended error handling facility will be added to the Mod II Library. The ADDNTRY operand indicates the number of user-defined and detected error conditions that are to be added to the standard extended error handling facility option table.

If all the operand defaults are desired, code the FORTLIB macro instruction without specifying any operands, as follows:

```
FORTLIB
```

In this case, all the underlined values in Figure 9 will be established as defaults for the library routines. The following paragraphs describe in detail the FORTLIB macro keyword operands and their respective parameters.

Code the FORTLIB macro instruction specifying any of the keyword operands listed in Figure 9.

Name	Operation	Keyword Operands and Parameters
	FORTLIB	UNTABLE= { <u>08</u> } { number } ONLNRD= { <u>05</u> } { unit } OBJERR= { <u>06</u> } { unit } ONLNPCH= { <u>07</u> } { unit } EOUNDRY= { <u>ALIGN</u> } { NOALIGN } OPTERR= { <u>EXCLUDE</u> } { INCLUDE } ADDNTRY= { <u>0</u> } { number }
<p>Note: All operands are optional; where a list of parameters is shown within braces, only one may be chosen. The underlined parameters are the defaults. Any number of operands may be chosen. When coding, column 1 must be blank. The FORTLIB may appear anywhere before column 72, but it must precede the operands by at least one blank. The operands are separated by commas and may be continued on any number of cards as long as column 72 contains a nonblank character and the following cards begin in column 16.</p>		

Figure 9. Format of the FORTLIB Macro Instruction

The FORTLIB macro instruction keyword operands are described in the following paragraphs.

UNTABLE=number

specifies the number of FORTRAN logical input/output units to be used at object-time. This number does not necessarily correspond to the number of input/output devices available to the installation's System/360 or System/370 machines.

number

is a two-digit integer from 08 to 99.

If the UNTABLE operand is not coded, a value of 08 is assumed.

ONLNRD

specifies which FORTRAN logical input/output unit is to be used when the READ (on-line) statement is encountered in a FORTRAN source program. The number specified must not exceed the value specified for the UNTABLE operand, and cannot be the same as the value specified for the OBJERR and ONLNPCH operands.

unit

is a two-digit decimal number from 01 to 99.

If the ONLNRD operand is not coded, a value of 05 is assumed.

OBJERR

specifies which FORTRAN logical input/output unit is to be used for object-time error messages and FORTRAN dumps. The number specified cannot exceed the value specified for the UNTABLE operand, and cannot be the same as the values specified for the ONLNRD and ONLNPCH operands.

unit

is a two-digit decimal number from 01 to 99.

If the OBJERR operand is not coded, a value of 06 is assumed.

ONLNPCH

specifies which FORTRAN logical input/output unit is to be used when the PUNCH (on-line) statement is encountered in a FORTRAN source program. The number specified cannot exceed the value specified for the UNTABLE operand, cannot be the same as the values specified for the OBJERR and ONLNRD operands.

unit

is a two-digit decimal number from 01 to 99.

If the ONLNPCH operand is not coded, a value of 07 is assumed.

BOUNDRY

controls the inclusion of the execution-time boundary alignment routine in SYS1.LINKLIB.

ALIGN

specifies that the routine is included.

NOALIGN

specifies that the routine is not included.

If the BOUNDRY operand is not coded, ALIGN is assumed. The NOALIGN operand should be specified for System/360 Model 85 and 195 machines. Under CMS, the default is NOALIGN.

OPTERR

controls the type of object-time error message facility to be generated.

EXCLUDE

specifies that minimum error service and the termination of the job when an error is detected.

INCLUDE

specifies that complete error service and continuation of the job after an object-time error is detected (dynamic and default control is provided for each error condition).

If the OPTERR operand is not coded, EXCLUDE is assumed.

ADDNTRY

specifies the number of option table entries to be generated in addition to the required number. The option table for the Extended Error Handling facility has an 8-byte entry for each IBM-designated error condition. An additional 8-byte entry should be added for each additional error condition to be installed. This operand is valid if, and only if, OPTERR=INCLUDE is specified.

number

is a decimal number from 000 to 599.

If the ADDNTRY operand is not coded, a value of 000 is assumed.

LOADING AND RUNNING THE USER-WRITTEN PROCEDURE AND THE LIBRARY DISTRIBUTION TAPE UNDER OS

To run the library distribution tape you must do the following:

- ① Execute the user-written procedure to prepare the system for loading the distribution tape. Running the procedure accomplishes the following:
 - Places the JCL procedure PPMODTWO into SYS1.PROCLIB. PPMODTWO identifies the libraries that will contain the Mod II Library modules (SYS1.FORTLIB and SYS1.LINKLIB or private libraries).
 - Places the FORTLIB macro instruction into SYS1.PPOPTION.
- ② Issue a START RDR,cuu to load the tape, where cuu specifies the channel and unit address of the tape device containing the distribution tape. Running the library distribution tape accomplishes the following:
 - Checks the coding of the FORTLIB macro instruction for valid syntax. See "Appendix C: Installation Messages" for diagnostic messages.
 - Assembles the library modules that depend upon the FORTLIB macro instruction.
 - Link edits the appropriate library modules into SYS1.LINKLIB or SYS1.FORTLIB or private subprogram and linkage libraries.
 - Punches a card deck that contains two programs. The first program is a sample program that is used to test the success of the library installation. The second program is used to move user-written subprograms from an old FORTLIB data set to the new FORTLIB data set created by the installation procedure.

Figure 10 describes the effects of the two steps listed above. The numbers in the figure correspond to the numbered steps.

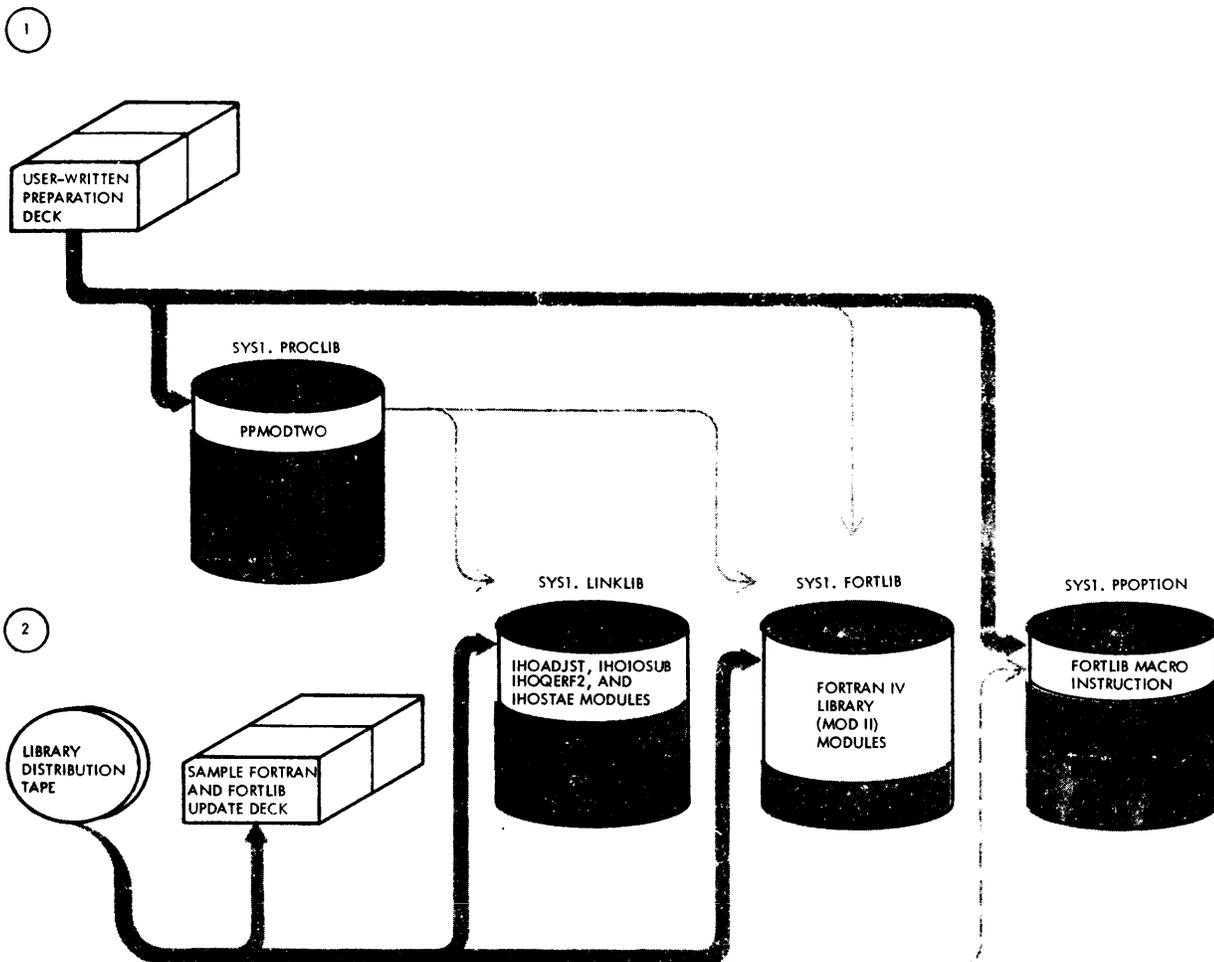


Figure 10. Effect of Running the User-written Procedure and the Library Distribution Tape Under OS

RUNNING A SAMPLE FORTRAN PROGRAM TO VERIFY THE SUCCESS OF THE LIBRARY INSTALLATION AND UPDATING SYS1.FORTLIB UNDER OS

The procedure on the distribution tape punches a card deck that contains two jobs.

1. The first job compiles, using the FORTRAN IV (H Extended) compiler, link edits, and executes a sample FORTRAN program.

If your SYS1.PROCLIB data set does not contain the FORTXCLG procedure, or if you wish to use a different compiler, you must modify the // EXEC card accordingly. In addition, ensure that the procedure's SYSLIB DD card refers to the data-set name of the FORTRAN IV Library (Mod II) just installed.

2. The second job may be used to move user-written programs from the old FORTLIB data set, if present, to the new subprogram library data set. The card deck for this job must be modified before execution, as follows:

- The UNIT= and VOL=SER= parameters must be supplied on the //SYSUT2 DD card.
- The UNIT= and VOL=SER= parameters must be supplied on the DD1 and DD2 DD cards for the unit and volume of the old and new subprogram library data sets.
- The COPY PDS control statement must be repunched specifying the following operands:

PDS = name of the old subprogram library data set
TO = device of the new subprogram library data set =
volume serial number of new subprogram library data
set
FROM = device of the old subprogram library data set =
volume serial number of the old subprogram data set
RENAME = name of the new subprogram library data set

Note: The COPY PDS control statement in this deck is followed by EXCLUDE statements assure that error messages will not be printed for each module that already exists in the newly created FORTRAN library.

FORTRAN IV LIBRARY (MOD II) UNDER VM/370

REQUIREMENTS FOR LIBRARY INSTALLATION UNDER CMS

For Library and PTF Installation

- A System/370 machine that can support VM/370.
- An installed release of VM/370.
- A minimum CMS configuration.
- The distribution tape for the FORTRAN IV Library (Mod II).
- The CMS library CMSLIB containing the extended arithmetic simulator routines, IEAXPSIM, IEAXPALL, and IEAXPDXR.

For Library Installation Only

- 270 blocks of unused space on your system (S) disk for the library (either a 2314 or the equivalent space on a 3330).
- Ten (10) cylinders of unused space on a 2314 or the equivalent space on a 3330 for a scratch system disk.

For PTF Installation Only

- The following CMS files that were installed by the library distribution tape when the library was originally installed:

(IHOCMSFT)	TXTLIB
IHOCMSFT	MACLIB
IHOSTAL	EXEC
IHOLIGEN	ASSEMBLE
IHOCMS	TEXT
IHOSAMP	TEXT
IHOCMS	ASSEMBLF
IHOCMSYE	TEXT
IHOCMSNO	TEXT

Note: The above files must be on disks that have been made available through the ACCESS command. The file named IHOCMSFT TXTLIB is produced by the installation procedure and given a name that you supply. You must rename this file back to IHOCMSFT TXTLIB before the library can be rebuilt. In addition, the OS Sample program with its JCL will be read into the scratch A disk as IHOCMSFT FORTRAN.

INSTALLING THE LIBRARY UNDER CMS

The compiler installation under CMS consists of the following steps:

- ① Mount the library distribution tape on virtual unit 181.
- ② Log into VM/370 and IPL CMS.
- ③ Issue an ACCESS command assigning the disk that you have selected as the scratch A disk.

- ④ Write and file the following EXEC procedure:

```
|edit prime exec
|EDIT;
|input
|INPUT;
|&control cms time
|tape rew
|&error &goto -errt
|tape fsf
|filedef inmove tapl (recfm fb lrecl 80 block 80 change)
|filedef outmove disk cmsfort data (recfm f lrecl 80 block 80
|change)
|&error &goto -err
|movefile
|cp spool punch to *
|punch cmsfort data (noheader
|read * *
|&begtype
|distribution tape is positioned beyond cmsfile ready to issue
|ihostal
|&end
|cp spool punch off
|&exit
|-errt &type tape move error error exit
|&exit2
|-err &type error in move file exit
|&exit1
|Ⓢ
|EDIT:
|file
```

Note: It is advisable to keep this procedure permanently, in the event another installation is to be performed at a future date.

- ⑤ Type in PRIME to execute the PRIME procedure, which reads file 2 from the installation tape and makes the following CMS files available:

<u>Filename</u>	<u>Filetype</u>	<u>Contents</u>
IHOSTAL	EXEC	Installation control procedure
IHOSAMP	FORTRAN	Sample program for CMS
IHOLIGEN	ASSEMBLE	Source for the FORTLIB macro instruction
IHOCMS	ASSEMBLE	Source for the compiler CMS interface routine

- ⑥ Type in IHOSTAL to execute the IHOSTAL procedure, which will issue a QUERY command for your terminal and print the following message:

```
|OPTION TO INSTALL OR REBUILD
|DO YOU HAVE ALL NECESSARY FILES ON DISK?
|IF RESPONSE IS 'YES' REBUILD
|IF RESPONSE IS 'NO' READ INSTALLATION TAPE
```

- ⑦ If this is to be an installation, enter no. If you are installing PTFs, enter yes and skip to step ⑩ .

Note: The alternative response of yes is for use when PTFs are to be applied to the library or system. When you receive a PTF you must remove the OS JCL either by punching out the PTF tape or by reading the tape into a CMS file and using the CMS editor. You must then insert the change into the appropriate TXTLIB for use by the installation procedure when rebuilding the compiler. For ease in segregating text decks for PTF installation, you will be requested in step ⑰ to name two additional text libraries, which will be placed in front of IHOCMSFT TXTLIB with a GLOBAL command by the installation procedure.

- ① As a check, the IHOSTAL procedure will type the following message at your terminal:

```
| IS FORTRAN DISTRIBUTION TAPE MOUNTED ON VIRTUAL TAPE UNIT 181?
| HAS THE CMS FILE BEEN READ OFF THE DISTRIBUTION TAPE?
| IS THE DISTRIBUTION TAPE POSITIONED AFTER CMS FILE?
| INSTALLATION OF 5734 LM3 LIBRARY MOD 2
| WHEN READY ENTER 'GO' OTHERWISE ENTER 'NOGO'
```

If the answer to the above questions is yes, enter go. If you have not done the items listed, enter nogo; do what is required, and re-execute the IHOSTAL procedure.

- ① The installation procedure offers you the opportunity to generate the unit assignment table with default record characteristics in a CMS format. The following message is typed out:

```
| DO YOU WISH CMS FILE CHARACTERISTICS
| IN UNIT ASSIGNMENT TABLE?
| THAT IS RECFM F BLOCK 80
| RESPOND 'YES' OR 'NO'
```

If you respond yes, CMS default characteristics will be established for all data set reference numbers except for those assigned to the printer, card punch, and card reader. The CMS characteristics are:

```
RECFM  F
LRECL  80
BLOCK  80
BUFNO  1
```

If you respond no, the supplied characteristics will be established. They are:

```
RECFM  U
LRECL  800
BLOCK  32768
BUFNO  2
```

Note: Records that are not defined as RECFM F, LRECL 80, and BLOCK 80 cannot be used with the CMS editor.

10

During the installation of the library, the IHOSTAL procedure types the following message at your terminal:

```
-----  
|DO YOU WISH TO ASSEMBLE INTERFACE  
|RESPOND 'YES' OR 'NO'  
-----
```

If you want the interface assembled, enter yes; the interface routine will be assembled, an IHOCMS TEXT file will be created, and a listing printed. If you answer no, the procedure checks to see that the library interface routine, IHOCMS TEXT, is available. If it is not, the interface routine will be assembled; if it is available, no new IHOCMS TEXT file or listing will be produced.

During the installation of a PTF, this check is omitted and the message, shown above, does not appear.

Note: If you wish to force the assembly of the interface routine you must first erase an existing IHOCMS TEXT file.

11

The IHOSTAL procedure will continue processing and type the following message at your terminal:

```
-----  
|IF IHOLIGEN ASSEMBLE IS AS DESIRED FOR OPTIONS ENTER 'NOEDIT'  
|IF YOU WISH TO EDIT THE OPTIONS STATEMENT THAT EXISTS ENTER  
|'EDIT'  
|TO EXIT THE CMS EDITOR YOU MUST ENTER CARRIAGE RETURN AND 'FILE'  
|AFTER THE OPTIONS HAVE BEEN ENTERED  
|TO EXIT INSTALL PROCEDURE IN CASE OF ERROR TYPE IN 'HALT'  
-----
```

See the section "Coding the FORTLIB Macro Instruction" for an explanation of the IBM-supplied default options. If all the defaults shown there are acceptable without any changes, type noedit and skip to step 13. If you want to change any of the IBM supplied defaults, type edit and go on to step 12.

12

The IHOSTAL procedure will type the following list of IBM-supplied compiler option defaults:

```
-----  
* MOD 2 FORTRAN LIBRARY OPTIONS DEFAULTS SHOWN  
FORTLIB UNTABLE=08, OBJERR=06, CNLMRD=05, CNLNPCH=07, *  
BOUNDARY=NOALIGN, OPTERR=EXCLUDE, ADDNTRY=0  
* ENTER MOD 2 OPTIONS DESIRED  
-----
```

To change any of the defaults, type in the macro, following the conventions for editing a file with a filetype of ASSEMBLE:

- A** Hit the **TAB** key once to position the carrier at the beginning of the operand field.
- B** Type in the macro name and each keyword operand to be changed together with its new default value.
- C** If the number of options to be changed requires a continuation line, type in a character as a continuation indicator in column 72 (count from the left hand margin indicator), and continue

typing the options beginning in column 16 of the following line.

- D** After all changes have been made, hit the RETURN key **CR** twice and issue the FILE subcommand.

Example:

```
|TAB| fortlib onlnpch=08  
|CR|  
|EDIT:  
|file
```

Note: You need not specify a value for the ADDNTRY and OPTERR operands. You will be specifically requested to enter parameters for them in steps **13** and **16**.

- 13** To obtain a value for the ADDNTRY operand of the FORTLIB macro instruction, the installation procedure types out the following message:

```
|ENTER NUMBER OF USFR ENTRIES IN OPTION TABLE  
|ZERO '0' FOR NO ENTRIES TO 598 MAXIMUM
```

You respond by supplying the number of user entries that you wish installed in your option table. If you will not be using the user-defined entries for the extended error handling facility, enter 0.

- 14** The procedure continues processing and types out the following message requesting you to name the TXTLIBS you want installed:

```
|ENTRY NAMES FOR THE TEXTLIB YOU WISH TO USE FOR THIS LIBRARY  
|YOU MAY ENTER 1, 2 OR 3 NAMES YOU MUST SUPPLY ONE NAME AT LEAST  
|NAME 1 IS FOR THE MAIN TXTLIB FOR FORTRAN OBJECT TIME LIBRARY  
|IF ONLY 2 NAMES ARE SPECIFIED THE PRIMARY CHOICE GOES TO NAME 1  
|THE SECONDARY CHOICE GOES TO NAME 2 TXTLIB  
|NAME 2 IS FOR A TXTLIB FOR ONLY EEH ROUTINES  
|NAME 3 IS FOR A TXTLIB FOR ONLY NOEEH ROUTINES
```

You can create up to three TXTLIBS and vary their contents depending on your needs, as follows:

- One Library (FORTMOD2)

If you are not using the extended error handling facility, your library will contain:

FORTMOD2 - files 4 and 5 of the distribution tape, IHOCMS, IHOUATBL, and IHOUOPT without error handling.

If you are using the extended error handling facility, your library will contain:

FORTMOD2 - files 4 and 6 of the distribution tape, IHOCMS, IHOUATBL, and IHOUOPT with error handling.

- Two Libraries (FORTMOD2 and MOD2NEEH or MOD2EEH)

If you are not using the extended error handling facility, your libraries will contain:

FORTMOD2 - file 4 of the distribution tape, IHOCMS, IHOUATBL, and IHOUOPT

MOD2NEEH - file 5 of the distribution tape and IHOUOPT

If you are using the extended error handling facility, your libraries will contain:

FORTMOD2 - files 4 and 5 of the distribution tape, IHOCMS, IHOUATBL, and IHOUOPT

MOD2EEH - files 4 and 6 of the distribution tape and IHOUOPT

- Three Libraries (FORTMOD2, MOD2EEH, and MOD2NEEH)

You may create three TXLIBs if you are using the extended error handling facility. Your libraries will contain:

FORTMOD2 - file 4 of the distribution tape, IHOCMS, and IHOUATBL

MOD2EEH - file 6 of the distribution tape and IHOUOPT (with extended error handling modules)

MOD2NEEH - file 5 of the distribution tape and IHOUOPT (without extended error handling modules)

⑮ Enter the library names you have chosen. (The names shown above are only suggestions. You may use them or your own names.)

⑯ To obtain a value for the OPTERR operand of the FORTLIB macro instruction, the installation procedure types out the following message:

```

FOR TXLIB BUILD INDICATE
'INCLUDE' FOR EXTENDED ERROR HANDLING AS PRIMARY CHOICE
'EXCLUDE' FOR NO EXTENDED ERROR HANDLING PRIMARY CHOICE
  
```

This message appears only if you have chosen 2 TXLIB names. If you want to install the extended error handling facility in your library, as your primary choice, enter include; if you do not, enter exclude.

⑰ To provide a facility for segregating text decks (refer to step ⑦ for more information) during the installation of PTFs, the installation procedure will type the following message at your terminal:

```

ENTER CR 1 OR 2 MACLIB NAMES
  
```

If you do not want any additional MACLIBs, hit the RETURN key, **CR**. If you want additional MACLIBs enter one or two names that will be used as the filename for the additional MACLIBs. They will

be placed before the IHOCMSFT MACLIB with a GLOBAL command issued by the installation procedure.

⑮ The procedure will continue processing until the installation of the library and interface on the scratch disk is complete.

⑯ At this point procedure will execute the library sample program, IHOSAMP TEXT, and type the following instructions at the terminal:

```
|RUN SAMPLE JOB TAKE CONSOLE PROMPTS  
|EXECUTION BEGINS...  
|SAMPLE JOB FOR MOD 2 LIBRARY  
|ENTER SHORT MESSAGE
```

Respond by typing in any short message, preceding the message with a blank or the character '1' in the first position as the carriage control character

Example:

```
|1short message
```

The procedure responds by repeating the first 11 characters of the message (beginning with the second position), followed by a set of statements indicating successful installation of the library:

```
|SHORT MESSA  
|END OF MOD FORTRAN LIBRARY SAMPLE JOB  
|END OF SAMPLE JOB  
|INSTALL OF MOD 2 FORTRAN LIBRARY COMPLETE  
|R;
```

⑰ To move the library from the scratch A disk to your actual system disk, issue the following commands assuming that 191 is the scratch disk and 190 is the system disk:

```
|access 191 b  
|access 190 a  
A |copy filename txtlib b2 = = a2  
B |copy filename txtlib b2 = = a2  
C |copy filename txtlib b2 = = a2
```

In the set of commands shown above, the lettered commands require that you supply information that is dependent upon your system. An explanation of the information required follows:

A You must include the first file name that you chose in step ⑮ and the same mode that you supplied in the command lettered **A** above.

- B** Enter this command if you specified a second name in step ⑩
- C** Enter this command if you specified a third name in step ⑩

Note: You may save the contents of the scratch disk for your archives or to rebuild the library at a later date.

STORAGE REQUIREMENTS

The storage estimate tables in this section provide the following:

- The amount of dynamic storage required for execution of the FORTRAN IV (H Extended) compiler under OS and CMS (see Table 1).
- The amount of secondary storage required by the FORTRAN IV (H Extended) compiler under OS (see Table 2).
- The amount of secondary storage required by the FORTRAN IV Library (Mod II) under OS (see Table 3).
- The amount of secondary storage required by the FORTRAN IV (H Extended) Compiler and Library (Mod II) under CMS.

Table 1. Minimum Dynamic Storage Requirements for Executing the FORTRAN IV (H Extended) Compiler Under OS and CMS

Control Program	Access Method	Storage Required (in bytes)
MFT	QSAM	160K
MVT	QSAM	160K
CMS	QSAM ¹	600K

¹CMS simulates the OS Queued Sequential Access Method.

Table 2. FORTRAN IV (H Extended) Compiler Secondary Storage Requirements

Description	Number of Directory Records ¹	Number of Tracks Required						
		IBM 2301 Drum	IBM 2302 Disk	IBM 2303 Drum	IBM 2311 Disk	IBM 2314 Disk	IBM 2321 Data Cell	IBM 3330 Disk
SYS1.PROCLIB	2	1	3	3	4	2	8	1
SYS1.LINKLIB	2	33	132	135	180	90	329	45
SYS1.PPOPTION	1	1	2	2	2	1	4	1

¹The number of 256-byte records allocated for a directory when a new partitioned data set is being defined. (See the description of the SPACE parameter of the DD statement in the publication IBM System/360 Operating System: Job Control Language Reference, Order No. GC28-6704.) The number of directory records that can be contained on a track is as follows:

- IBM 2301 Drum Storage - 45
- IBM 2302 Disk Storage - 14
- IBM 2303 Drum Storage - 12
- IBM 2311 Disk Storage - 10
- IBM 2314 Disk Storage - 17
- IBM 2321 Data Cell - 5
- IBM 3330 Disk Storage - 29

Table 3. FORTRAN IV Library (Mod II) Secondary Storage Requirements

Description	Number of Directory Records ¹	Number of Tracks Required						
		IBM 2301 Drum	IBM 2302 Disk	IBM 2303 Drum	IBM 2311 Disk	IBM 2314 Disk	IBM 2321 Data Cell	IBM 3330 Disk
SYS1.FORTLIB	58	12	47	48	65	31	117	21
SYS1.LINKLIB	2	1	2	2	3	2	5	1
SYS1.PPCPTION	1	1	2	2	2	1	4	1

¹The number of 256-byte records allocated for a directory when a new partitioned data set is being defined. (See the description of the SPACE parameter of the DD statement in the publication IBM System/360 Operating System: Job Control Language Reference, Order No. GC28-6704.) The number of directory records that can be contained on a track is as follows:

- IBM 2301 Drum Storage - 45
- IBM 2302 Disk Storage - 14
- IBM 2303 Drum Storage - 12
- IBM 2311 Disk Storage - 10
- IBM 2314 Disk Storage - 17
- IBM 2321 Data Cell - 5
- IBM 3330 Disk Storage - 29

**Table 4. FORTRAN IV (H Extended) Compiler and Library (Mod II)
Secondary Storage Requirements under CMS**

Description	Number of Cylinders Required for Installation		Number of Blocks Required
	IBM 2314 Disk	IBM 3330 Disk	
H Extended Compiler	30	15	542
Library (Mod II)	10	5	270

SYSTEM PROGRAMMING

COMPILER AND LIBRARY INSTALLATION FOR OS RELEASE 19 USERS

REQUIREMENTS FOR INSTALLATION ON OS RELEASE 19

To install the FORTRAN IV (H Extended) compiler and the FORTRAN IV Library (Mod II) on Release 19 of the System/360 Operating System, you must have the following available:

For both compiler and library

- A System/360 or System/370 machine that supports the MFT (with the ATTACH option) and/or MVT System/360 Operating System environment(s).
- The Distribution Library for Release 20 of the IBM System/360 Operating System.

Note: The distribution library must be restored from its original tape to disk device before it can be used. The restore procedure assigns the data-set name and the volume serial number to be used. See the publication IBM System/360 Operating System: Utilities, Order No. GC2²-6586 for information on the restore procedure.

- 136K bytes of main storage for processing each distribution tape.
- IMASPZAP fixes for APAR's 31594, 34556, and 39145 and Program Temporary Fix (PTF) for APAR 34018.
- The IEBCOPY, IEBUPDTE, and IEHPROGM utility programs.
- The IMASPZAP service aid program.

For the compiler only

- Space available on SYS1.LINKLIB or a private library for the FORTRAN IV (H Extended) compiler (see Table 2 in the "Storage Estimates" section for SYS1.LINKLIB storage requirements).
- Space available on SYS1.PROCLIB or a private library for the FORTRAN IV (H Extended) cataloged procedures (see Table 2 in the "Storage Estimates" section for SYS1.PROCLIB storage requirements).

For the library only

- Space available on SYS1.LINKLIB or a private library for the IHOSTAE, IHOIOSUB, and IHOQERF2 modules.

Note: If boundary alignment is requested in the FORTLIB macro instruction, the IHOADJST module is included on SYS1.LINKLIB or a private library. See Table 3 in the "Storage Estimates" section for the storage required by SYS1.LINKLIB.

- Space available for the SYS1.FORTLIB data set or a private library, which will contain the remainder of the Mod II Library modules (see Table 3 in the "Storage Estimates" section for SYS1.FORTLIB storage requirements).

- Additional space on SYS1.LINKLIB or a private library for the Extended Precision Floating Point Simulator routines. If you are using a private library, you must use a JOBLIB DE statement that defines the data set on which the library resides each time you run a FORTRAN IV (H Extended) program.
- 130 queue records for the system initiator/terminator.

Note: The number of queue records is determined at system generation time by the JOBQLMT parameter in the SCHEDULR macro instruction. If the value specified for JOBQLMT was less than 130, it must be raised to 130 during an initial program load (IPL) prior to installing the library. During the IPL, the operator must reply to the message:

id IEF423A SPECIFY JOB QUEUE PARAMETERS

with the following:

REPLY id,',130'

This reply is given in addition to any others that may be requested.

INSTALLATION PROCEDURES ON OS RELEASE 19

To install the FORTRAN IV (H Extended) compiler and the FORTRAN IV Library (Mod II) on Release 19, do the following:

- Modify Release 19 of the IBM System/360 Operating System as described in the section "Copying the Extended Precision Floating Point Simulator Routines."
- Using the IMASPZAP service aid program, apply fixes for APAR's 31594, 34556, and 39145 and apply Program Temporary Fix (PTF) 70260 for APAR 34018.
- Follow the installation procedures for Release 20.

Note: If you are using an IBM System/360 Model 85 or 195, or a System/370 that has the Extended Precision Floating Point Arithmetic feature, see the section "Special Considerations for System/360 Model 85 or 195, or System/370 Users" for additional processing that is required.

Copying the Extended Precision Floating Point Simulator Routines

Prepare a card deck to copy the Extended Precision Floating Point Simulator routines from the Release 20 distribution library into SYS1.LINKLIB, or a private linkage library. These routines are not part of Release 19 but are required by the FORTRAN IV (H Extended) compiler and the FORTRAN IV Library (Mod II). Figure 11 shows the JCL required. The lettered statements contain items that you must provide based on the facilities available to you.

	//STEP1	EXEC	PGM=IEBCOPY
	//SYSPRINT	DD	SYSOUT=A
A	→ //DD2	DD	DSN=SYS1.CI505,DISP=OLD,
B	→ //		UNIT=unit,VOL=SER=volume serial,
			SPACE=(TRK,(1,1))
C	→ //DD3	DD	DSN=linkage library name,DISP=OLD
	//SYSUT3	DD	DSN=TEMP1,DISP=(NEW,DELETE),UNIT=unit,
	//		VOL=SER=volume serial,SPACE=(TRK,(1,1))
D	→ //SYSUT4	DD	DSN=TEMP2,DISP=(NEW,DELETE),UNIT=unit
	//		VOL=SER=volume serial,SPACE=(TRK,(1,1))
	//SYSIN	DD	*
		COPY	OUTDD=DD3,INDD=DD2
		SELECT	MEMBER=((IEAXPALL,,R),(IEAXPDXR,,R),
	//		(IEAXPSIM,,R))
	//		
	/*		

Figure 11. Copying the Extended Precision Floating Point Simulator Routines from Release 20 into Release 19

An explanation of the lettered items in Figure 11 follows:

A If you are using a 2311 Disk Storage device for the disk pack that contains the Release 20 starter system data set (SYS1.CI505), code 2311 for the unit parameter and DLIB03 for the volume serial number. If you are using a 2314 Disk Storage device, code 2314 for the unit parameter and DLIB01 for the volume serial number.

B Insert the name of the linkage library on which the Extended Precision Floating Point Simulator routines will reside (either SYS1.LINKLIB or a private library). If the library you choose is not cataloged, add to this statement the unit and volume serial number on which it resides.

Note: If you are using a private library, you may concatenate it with SYS1.LINKLIB at initial program load (IPL) time or you may include in the JCL of any FORTRAN program that will use the compiler or MOD II Library, a JOBLIB or STEPLIB DD statement that describes the private library on which the simulator routines reside. See the "Link Library List" section for information on concatenating data sets with SYS1.LINKLIB or the publication IBM System/360 Operating System Job Control Language Reference, Order No. GC28-6704, for information on using the JOBLIB or STEPLIB statements.

C Insert the type of the direct-access device and the volume serial number you are using for the work data set.

D Insert the type of the direct-access device and the volume serial number you are using for the work data set.

After modifying Release 19 of the operating system, continue installation processing using the information contained in the sections following the installation requirements section for either the compiler or the library. The two-steps described above may be run with the user-written procedures discussed in those sections; however, they must be run before you attempt to run the sample FORTRAN program deck.

SPECIAL INSTALLATION CONSIDERATIONS FOR SYSTEM/360 MODEL 85 OR 195, OR SYSTEM/370 USERS WITH OS

If you have a System/360 Model 85 or 195 or a System/370 that has the Extended Precision Floating Point Arithmetic feature, your system is able to automatically perform extended precision arithmetic operations using this built-in hardware. Extended precision division operations, however, require a simulator routine that assists the hardware and increases the speed of the operations. There are two routines available. Your choice of the appropriate routine to use depends upon your hardware. Using the right routine will increase the speed of extended-precision division operations on systems equipped with extended-precision floating-point arithmetic.

The Release 20 distribution library contains both the division simulator routine and a complete simulator routine for users that do not have the Extended Precision Floating Point Arithmetic feature. If you wish to take advantage of this feature, you must indicate to the system that it is available or the complete simulator routine will be used. To indicate that the Extended Precision Floating Point Arithmetic feature is available and the division simulator routine is to be used, you must modify bit 7 of byte 182 in the Communications Vector Table. The following procedures perform the necessary modifications.

1. Prepare a card deck that uses the IMASPZAP service aid program to dump the CSECT, IEAQBK00 for MVT and CSECT, IEACVTRN for MFT in the module IEANUC01 of nucleus that you are using. If you have more than one nucleus available, specify the same CSECT in the corresponding module of the nucleus you wish to modify (IEANUC02, etc.). You may run this step with the user-written procedure described in the "Installation Procedures" section. For MVT systems, the card deck should contain the following:

```
//STEP3      EXEC  PGM=IMASPZAP
//SYSLIB     DD    DSN=SYS1.NUCLEUS,DISP=OLD
//SYSPRINT  DD    SYSOUT=A
//SYSABEND  DD    SYSOUT=A
//SYSIN     DD    *
              DUMP IEANUC01  IEAQBK00
/*
```

For MFT systems, replace the DUMP statement with DUMP IEANVC01 IEACVTRN

2. Using the dump produced above, locate the contents of byte 182 (decimal) in the Communications Vector Table (CVT). This byte is located at a displacement of 182 bytes beyond IEACVT, an entry point in IEAQBK00 and IEACVTRN. Calculate the location of this byte using the system generation assembler listings for the nucleus you are using (IEANUC01, IEANUC02, etc.).

3. Prepare a second card deck that will replace the contents of bit 7 of byte 182 with a bit that is set to one. All the other bit settings of the byte must remain unchanged. For MVT systems, the second deck should contain the following:

```

//JOB2      JOB      accounting information, MSGLEVEL=(1,1)
//STEP      EXEC     PGM=IMASPZAP
//SYSLIB    DD       DSN=SYS1.NUCLEUS, DISP=OLD
//SYSPRINT  DD       SYSOUT=A
//SYSABEND  DD       SYSOUT=A
//SYSIN     DD       *
              NAME    nucleus data set name IEAQBK00
              VERIFY  location of byte 182  contents of byte 182
              REP     location of byte 182  replacement for
                                      byte 182

```

For MFT systems, replace the NAME statement with NAME nucleus data set name IEACVTRN

4. Complete the remaining steps necessary to install on Release 19.

ASSEMBLING USER-WRITTEN COMPILER AND LIBRARY MODULES ON OS RELEASE 19

To assemble on Release 19 compiler and library modules that you have written for use with the FORTRAN IV (H Extended) compiler or the FORTRAN IV Library (Mod II), you must use the Release 20 Assembler F and macro library. These two Release 20 components must be restored from their original distribution library tape to a disk device before they can be used. The restore procedure assigns the data-set name and volume serial number to be used. See the publication IBM System/360 Operating System Utilities, Order No. GC28-6586 for information on the restore procedure. You may assemble or reassemble modules following the installation process. But before you attempt to run a FORTRAN program that uses your modules; the following example shows how this might be done in a simple case.

```

//MYJOB      JOB      accounting information, MSGLEVEL=(1,1)
//JOBLIB     DD       DSN=SYS1.AS037, DISP=OLD, UNIT=2314,
//           VOL=SER=DLIB01
//MYSTEP     EXEC     ASMEC
//ASM.SYSLIB DD       DSN=SYS1.MACLIB, DISP=OLD, UNIT=2314,
//           VOL=SER=DLIB01
//ASM.SYSIN  DD       *
              .
              .
              .
              (your program source statements)
              .
              .
              .
/*

```

If you are using a 2311 Disk Storage device, the volume serial number for the Assembler F is DLIB05; the volume serial number for the macro library is DLIE01.

EXTENDED ERROR HANDLING FACILITY

HOW TO CREATE OR ALTER AN OPTION TABLE

The option table is supplied during system generation or installation of the library when the extended error handling facility is requested. However, to provide a new set of options for an existing table, the option table must be reassembled and link edited into the FORTRAN library. This is done after system generation or installation of the compiler and before you attempt to run a program that requires the extended error handling facility. A procedure for accomplishing this is described in the following text. A complete description of the option table, PREFACE option, and SETENT option can be found in the publication IBM System/360 Operating System: FORTRAN IV (H Extended) Programmer's Guide, Order No. SC28-6852.

An assembler language macro definition can be used to generate an option table. The macro definition and use of the macro for each option table entry are supplied as input to the assembler procedure ASMFCL to replace the system-generated option table with the new one.

An example of an assembler language macro definition used to generate an option table is shown in Figure 12. This example may be used as a guide by the user.

In the example, the macro parameters are as follows:

PREFACE a,b,c

- a
is the number of user entries to be created (same as the number specified for the ADDNTRY operand of the FORTLIB macro instruction).
- b
is the boundary alignment desired. A value of 0 is used for no alignment; a value of 1 is used for alignment.
- c
is the number of times the SETENT macro instruction (see below) is to be issued.

SETENT (a, b, c, d)

- a
is the error entry to be altered.
- b
is the count of errors to be allowed. A value of 0 indicates unlimited error occurrence.
- c
is the count of the number of times the message should be printed before suppression.
- d
is the hexadecimal digits that specify the option bits.

```

//OPTAB JOB 1,'SAMPLE MACRO',MSGLEVEL=1 CREATE IHOUOPT
//VER1 EXEC ASMFPC,PARM.ASM=NODECK
//ASM.SYSIN DD *
MACRO
  PREFACE %ADENT,%ADJUST,%SETENT
.* THIS MACRO GENERATES THE PREFACE TO THE OPTION TABLE AND SETS
.* GLOBALS FOR SUBSEQUENT CALLS TO THE SETENT MACRO
.* THE USE OF THIS MACRO GENERATES AN OPTION TABLE AS DEFINED BY IBM
.* AND ALLOWS CHANGES TO INDIVIDUAL ERROR NUMBERS AS DESIRED, BY USE
.* OF SETENT
  GBLA %COUNT,%TOTAL,%SETNR
  LCLA %A
IHOUOPT CSECT
%SETNR SETA %SETENT
%COUNT SETA 204 ERROR NUMBER OF FIRST ENTRY IN TABLE
%TOTAL SETA %ADENT+301 NUMBER OF LAST ENTRY IN TABLE
%A SETA %ADENT+98
  DC F'%A' TOTAL NUMBER OF ENTRIES IN TABLE
  DC B'0%ADJUST.000000'
  DC AL3(0)
MEND
MACRO
  SETENT %E
  GBLA %COUNT,%TOTAL,%SETNR
  LCLA %B
  %B SETA 1
%SETNR SETA %SETNR-1
.%AGAIN ANOP START OF LOOP TO GEN ONE ENTRY IN TABLE FOR ERROR NUMBER
  AIF (%COUNT GT %TOTAL).MEND HAVE ALL ENTRIES BEEN CREATED
  AIF (%B LE N'%SYSLIST').TEST
  AIF (%SETNR EQ 0).DEFAULT
  MEXIT
.%TEST ANOP
.* IF THERE IS NO USER SUPPLIED INFO FOR THIS ERROR NO TAKE DEFAULT
  AIF (%SYSLIST(%B,1) NE %COUNT).DEFAULT
ERR%COUNT DC AL1(%SYSLIST(%B,2)) NUMBER OF ERRORS TO ALLOW FR SETENT
  DC AL1(%SYSLIST(%B,3)) NO OF MSGS TO PRINT FROM SETENT
  DC X'00'
  DC X'%SYSLIST(%B,4)' OPTION BITS SUPPLIED BY SETENT
  DC F'1'
%COUNT SETA %COUNT+1
%B SETA %B+1
  AGO .AGAIN RETURN TO LOOP
.%DEFAULT ANOP IBM DEFAULTS FOR ERRORS NOT INDICATED BY SETENT
.* IBM SPECIAL CASES FOR MESSAGE COUNT
  AIF (%COUNT EQ 208).UNLIM
  AIF (%COUNT EQ 210).UNLIM
  AIF (%COUNT EQ 215).UNLIM
  AIF (%COUNT EQ 217).ONE
  AIF (%COUNT EQ 230).ONE
ERR%COUNT DC AL1(10)
.BACK1 ANOP
  DC AL1(5)
.BACK2 ANOP
  DC X'00'
.* IBM SPECIAL CASES FOR OPTION BITS
  AIF (%COUNT EQ 212).SPBITS
  AIF (%COUNT EQ 215).SPBITS
  AIF (%COUNT EQ 218).SPBITS
  AIF (%COUNT EQ 221).SPBITS
  AIF (%COUNT EQ 222).SPBITS
  AIF (%COUNT EQ 223).SPBITS
  AIF (%COUNT EQ 224).SPBITS
  AIF (%COUNT EQ 225).SPBITS
  DC X'42'
  AGO .CONT
.SPBITS DC X'52'
.CONT ANOP
  DC F'1'
%COUNT SETA %COUNT+1
  AGO .AGAIN RETURN TO LOOP
.UNLIM ANOP
ERR%COUNT DC AL1(0)
  AIF (%COUNT NE 210).BACK1
  DC AL1(10)
  AGO .BACK2
.ONE ANOP
ERR%COUNT DC AL1(1)
  DC AL1(1)
  AIF (%COUNT EQ 217).BACK2
  DC X'00'
  DC X'02'
  AGO .CONT
.MEND ANOP
  MEND
*
* END OF MACRO DEFINITION
*
* EXAMPLE OF THE USE OF THE MACRO
*
  PREFACE 50,1,2
  SETENT (220,5,2,21),(235,10,5,42),(255,2,0,4)
  SETENT (300,56,65,3)
END
/*
END OF DATA

```

Figure 12. Example of Assembler Language Macro Definition Used to Generate Option Table

The macro instructions are used as follows:

1. Only one PREFACE macro instruction is allowed.
2. As many SETENT macro instructions as are desired may be used. From 1 to 200 error entries can be specified in the use of a single SETENT macro instruction by using continuation cards.
3. Only error entries that differ from the default options need be specified. See the publication OS FORTRAN IV (H Extended) Programmer's Guide, Order No. SC28-6852, for more information on the default options.
4. Error codes must be placed in ascending order in the SETENT macro instruction. For IBM-supplied entries, error codes are in the range 302 to 899.
5. Changing one option for any error entry requires that all four parameters be specified. If default values are desired for an entry, they must be respecified. For example:

```
SETENT (241,50,5,42)
```

indicates that for error 241, the number of errors to be allowed is 50; the error message should be printed five times before it is suppressed; and the option bits are set with a hexadecimal 42.

THE LINK LIBRARY LIST UNDER OS

The link library list (LNKLST00) enables you to concatenate up to 16 data sets, on multiple volumes, to form SYS1.LINKLIB. When the system is generated, LNKLST00 is included in the system as a required member of SYS1.PARMLIB. If SYS1.PARMLIB does not include the member LNKLST00, SYS1.LINKLIB will be used as the system link library and a warning message will be provided.

LNKLST00 contains one member, SYS1.LINKLIB. After system generation, you will have the option of adding members via the IEBUPDTE utility program. Each member may have up to 16 extents. After making additions to SYS1.SVCLIB, SYS1.LINKLIB, or data sets concatenated to LINKLIB via LNKLST00, and before using the additions, an IPL should be performed to update the description of the link and/or SVC library in main storage.

Your input format (to IEBUPDTE) consists of 80 character records. Continuation is indicated by placing a comma after the last name in a record and a nonblank character in column 72. Subsequent records must start in column 16. The initial format is:

```
[b...] SYS1.LINKLIB
```

To add member names to LNKLST00, replace the initial record with:

```
[b...] SYS1.LINKLIB,name1,name2,name3,...
```

SYSLIN DEFINITION IN CATALOGED PROCEDURES

In order to allow the compiler to run more efficiently, cataloged procedures define SYSLIN as having a blocking factor of 40, i.e., a blocksize of 3200 bytes. Note that the blocking factor used during link editing bears a relationship to the linkage editor's SIZE option.

If the defaults at your installation do not handle the 40 to 1 blocking factor, change your cataloged procedure by either:

1. Specifying a blocksize of less than 3200 bytes for SYSLIN, or
2. changing the SIZE option and/or the REGION size used by the linkage editor. See the publication IBM System/360 OS Linkage Editor and Loader, Order No. GC28-6538, for further details.

Note also that the REGION size for the link edit step in the cataloged procedure is 96K.

APPENDIX A: COMPILER AND LIBRARY MODULES

FORTRAN IV (H EXTENDED) COMPILER MODULES

IFEAAA	IFECPX	IFEJGR	IFEQMT	IFETIS
IFEAAB	IFECSP	IFEKCN	IFEQPF	IFETLM
IFEAAC	IFECSR	IFEKOP	IFEQSM	IFETLS
IFEAAD	IFECTN	IFEKPA	IFEQSR	IFETMP
IFEAAE		IFEKRE	IFEQTL	IFETPK
IFEADC	IFEDCL	IFEKSA	IFEQWT	IFETPR
	IFEDIO	IFEKSM	IFEQXM	IFETRN
IFEAIB		IFEKST	IFEQXS	IFETSR
IFEAIO	IFEGAA	IFEKUN	IFEQXX	
IFEALD	IFEGCR			IFEUED
IFEANT	IFEGCZ	IFELAB	IFERBX	IFEUEN
IFEAPD	IFEGDA	IFELER	IFERBP	
IFEAPT	IFEGDF	IFELGN	IFERFL	IFEVCA
IFEATB	IFEGDT	IFELMA	IFERFP	IFEVCC
IFEATC	IFEGEV	IFELOK	IFERFR	IFEVFN
IFEATM	IFEGFM	IFELRG	IFERGB	
	IFEGNL	IFELTB	IFERLA	IFEWCN
IFEBUF	IFEGOS		IFERLL	
	IFEGSD	IFEPBD	IFERRG	IFEXRF
IFECAA	IFEGST	IFEPBL	IFERSL	IFEXRS
IFECAR		IFEPBT	IFERSS	
IFECAR	IFEJAB	IFEPBZ	IFERSX	IFEYER
IFECBR	IFEJAC	IFEPLP	IFERTB	
IFECDI	IFEJAL	IFEPLS		IFEZER
IFECDO	IFEJAN	IFEPOP	IFESBS	
IFECDP	IFEJBF			IHOFCONI
IFECDT	IFEJCM		IFETAA	IHOFTEN
IFECGC	IFEJCP	IFEQBM	IFETAB	IHOFTEN
IFECGO	IFEJDF	IFEQCF	IFETEN	
IFECGW	IFEJFI	IFEQCL	IFETEP	IHOQCONI
IFECIO	IFEJFU	IFEQKO	IFETIO	IHOCONO
IFECLT				

FORTRAN IV (H EXTENDED) COMPILER CMS INTERFACE MODULE

IFECMS

OBTAINING CSECT SIZES FOR THE COMPILER MODULES UNDER OS

It is possible to obtain CSECT sizes for the FORTRAN IV (H Extended) compiler modules using the IMBDMAP service aid program. A sample of the output from this program is shown in Figure 13. You need the following JCL statements:

```
//CSCTSIZE JOB accounting information, MSGLEVEL=(1,1)
//STEP1 EXEC PGM=IMBDMAP,,REGION=200K
//DD1 DD DSN=data-set name on which compiler
// resides (IFEAAB),DISP=SHR
//SYSPRINT DD SYSOUT=A
//
```

For more information on using the IMBDMAP service aid program see the publication IBM System/360 Operating System: Service Aids, Order No. GC28-6719.

FORTRAN IV LIBRARY (MOD II)

IHOADJST ^{1,2}	IHODEBUG	IHOEIFIX	IHOLASCN	IHOQSQRT
IHOASYNC ^{1,2}	IHOEIOSE**	IHOEFINTH**	IHOLATN2	IHOQTANH
		IHOEFIXPI	IHOLDFIG	IHOQTNCT
IHOEGOTO	IHOECOMH*	IHOEFIOSH**	IHOLERF	
IHOCLABS	IHOEDIOS*	IHOEFIOS2	IHOLEXP	IHOASASCN
IHOCLAS	IHOEFIOS*	IHOEFMAXD	IHOLGAMA	IHO SATAN
IHOCLEXP	IHOEFINTH*	IHOEFMAXI	IHO LLOG	IHO SATN2
IHOCLLOG	IHOERRM	IHOEFMAXR	IHO LSCN	IHO SERF
IHOCLSON	IHOETRCH*	IHOEFMODI	IHO LSCNH	IHO SEXP
IHOCLSQT		IHOEFMODR	IHO LSQRT	IHO SGAMA
IHO COMH2	IHOFAINT	IHOFOPT	IHO LTANH	IHO SLOG
IHO CQSCN	IHOFCDXI	IHOFOVER	IHO LTNCT	IHO SSCN
IHO CQSQT	IHOFCOMH**	IHOFOQTN		IHO SSCNH
IHO CQABS	IHOFCONI	IHOFOQXPI	IHO NAMEL	IHO SSQRT
IHO CQEXP	IHOFCONO	IHOFOXPQ		IHO STAE ²
IHO CQLOG	IHOFCQXI	IHOFRXPI	IHO QASCN	IHO STANH
IHO CQRIT	IHOFCVTH	IHOFRXPR	IHO QATN2	IHO STNCT
IHO CSABS	IHOFCXPI	IHOFSLIT	IHO QCONI	
IHO CSAS	IHOFDUMP	IHOFTEN	IHO QCONO	IHO TRCH**
IHO CSEXP	IHOFDVCH		IHO QERF	
IHO CSLOG	IHOFDXPD	IHOIBERH	IHO QERF ²	IHO TABL
IHO CSSCN	IHOFDXPI	IHOIBERR	IHO QSCN	IHO OPT
IHO CSCQT	IHOFEEXIT	IHOIOSUB ^{1,2}	IHO QSCNH	

¹Not available under CMS
²Available under OS on SYS1.LINKLIB
*Extended error handling routines
**Non-extended error handling routines

FORTRAN IV LIBRARY (MOD II) CMS INTERFACE MODULE

IHO CMS

The CSECT sizes for the FORTRAN IV Library (Mod II) modules can be found in the publications IBM OS FORTRAN IV Library - Mathematical and Service Subprograms, Order No. GC28-6818 and IBM OS FORTRAN IV Mathematical and Service Subprograms Supplement for Mod I and Mod II Libraries, Order No. SC28-6864.

APPENDIX B: SAMPLE PROGRAMS PRODUCED BY THE COMPILER AND LIBRARY DISTRIBUTION TAPES

FORTRAN IV (H EXTENDED) COMPILER UNDER OS

The sample program that is produced by the compiler distribution tape procedure is shown in Figure 14.

```

LEVEL 9J ( FEB 71 )                CS/360 FORTRAN H EXTENDED                CASE 71.057/10.23.41                PAGE 1
REQUESTED OPTICNS: MAP,LIST
OPTIONS IN EFFECT: NAME( MAIN),NOCOPTIMIZE,LIRECCUNT(60),SIZE(MAX),AUTOCBL(NONE),
SOURCE,EBCDIC,LIST,NOCHECK,CBJECT,MAP,NOFCRMAP,NOGCSTMT,NOXREF,NOALC,NCANSF,FLAG(1)

C GENERALIZED TEST CASE
C TEST ADP AND XP WITH NESTED DO'S, IF'S WITHIN DO'S, VARIABLE
C SUBSCRIPTS, GO TO WITHIN DO LOOP, MULTIPLE IF'S IN SUCCESSION.
C THE ROUTINE ITSELF APPROXIMATES THE SLOPE OF X**2 + X**3. THE TEST
C CASE IS SELF CHECKING. SLOPE IS CALCULATED AT THE 5 POINTS ON THE
C CURVE, 1.,2.,3.,4., AND 5.
ISN CCG2      REAL*8 S(50),T(50),W(50)
ISN CCG3      DO 25 I=1,6
ISN CCG4      1  E=C.COC1
ISN CCG5      2  TZ=I-1
ISN CCG6      3  P=(TZ+1.0)*TZ**2
ISN CCG7      4  CELT = 0.1
ISN CCG8      5  T (I) = TZ + CELT
ISN CCG9      6  S (I) = (T(I) + 1.0)*T(I)**2
ISN CCG10     7  DELS = S(I) - P
ISN CCG11     8  W(I) = DELS/DELT
ISN CCG12     9  DO 16 J =2,50
ISN CCG13    10  CELT = 0.1 * DELT
ISN CCG14    11  T(J) = TZ + CELT
ISN CCG15    12  S(J) = (T(J) + 1.0) *T(J)**2
ISN CCG16    13  DELS = S(J) - P
ISN CCG17    14  W(J) = DELS/DELT
ISN CCG18    19  IF (J - 2) 20,21,20
ISN CCG19    20  A = W(J-1) - W(J)
ISN CCG20    B = W(J-2) - W(J-1)
ISN CCG21    IF (A - B) 21,22,22
ISN CCG22    21  IF (W(J-1) - W(J) - E) 23,16,16
ISN CCG23    16  CONTINUE
ISN CCG24    22  V = W(J-1)
ISN CCG25    GO TO 24
ISN CCG26    23  V = W(J)
ISN CCG27    24  IF (TZ.EQ.C..AND.V-.GT..1) GO TO 26
ISN CCG28    IF (TZ.EQ.C.) GO TO 25
ISN CCG29    IF (TZ.EQ.1..AND.V-5..GT..1) GO TO 26
ISN CCG30    IF (TZ.EQ.1.) GO TO 25
ISN CCG31    IF (TZ.EQ.2..AND.V-16..GT..1) GO TO 26
ISN CCG32    IF (TZ.EQ.2.) GO TO 25
ISN CCG33    IF (TZ.EQ.3..AND.V-33..GT..1) GO TO 26
ISN CCG34    IF (TZ.EQ.3.) GO TO 25
ISN CCG35    IF (TZ.EQ.4..AND.V-56..GT..2) GO TO 26
ISN CCG36    IF (TZ.EQ.4.) GO TO 25
ISN CCG37    IF (TZ.EQ.5..AND.V-85..GT..2) GO TO 26
ISN CCG38    IF (TZ.EQ.5.) GO TO 25
ISN CCG39    GO TO 25
ISN CCG40    26  WRITE (6,100) TZ,V
ISN CCG41    25  CONTINUE
ISN CCG42    WRITE (6,101)
ISN CCG43    100  FORMAT (' TC11 FAILED WITH TZ AND V =, RESPECTIVELY,',F4.1,F12.4)
ISN CCG44    101  FORMAT (' TC11 COMPLETED')
ISN CCG45    STOP
ISN CCG46    END

```

Figure 14. Sample Program Produced by the Compiler Distribution Tape Under OS

FORTRAN IV (H EXTENDED) COMPILER UNDER CMS

The sample program that is produced by the compiler distribution tape procedure is shown in Figure 15. This source program listing is placed in the IFESAMP LISTING file produced by the compiler depending upon the options you specified in step 12 of the installation procedure.

```
LEVEL 2 ( SEPT 6 72 )                OS/360 FORTRAN H EXTENDED                DATE 72.271/13.01.31                PAGE 1
REQUESTED OPTIONS: SOURCE MAP LIST XREF TERM
OPTIONS IN EFFECT: NAME( MAIN),NOOPTIMIZE,LINCOUNT(60),SIZE(MAX),AUTODBL(NONE),
SOURCE,EBCDIC,LIST,NODECK,OBJECT,MAP,NOFORMAT,NOGOSTMT,XREF,NOALC,NOANSF,TERM,FLAG(S)
C IFESAMP FORTRAN H EXTENDED SAMPLE PROGRAM                                IFE00010
ISN 0002      C      WRITE(6,10)                                          IFE00020
ISN 0003     10     FORMAT(' SAMPLE JCB FOR H EXT COMPILER/' ' ENTER SHORT MESSAGE ') IFE00030
ISN 0004     10     READ (5,11)                                           IFE00040
ISN 0005     11     FORMAT ('TEXT ENTERED')                                IFE00050
ISN 0006     11     WRITE (6,11)                                           IFE0006C
ISN 0007     11     X=1.                                                   IFE00070
ISN 0008     11     WRITE (6,12)                                           IFE00080
ISN 0009     12     FORMAT(' END OF H EXTENDED SAMPLE JOB')              IFE00090
ISN 0010     12     STOP                                                    IFE0010C
ISN 0011     12     END                                                    IFE0011C
```

Figure 15. Sample Program Produced by the Compiler Distribution Tape Under CMS

REQUESTED OPTIONS: LIST,MAP

OPTIKAS IN EFFECT: NAME(MAIN),ACCP(IZIE,LINCCCLAT(60),SIZE(MAX),AUTCCBL(ACNE),
SOURCE,EBCDIC,LIST,NOCHECK,CBJECT,MAP,NOFORMAT,NOGSTMT,NOXREF,NOALC,NCANSF,FLAG(1)

```

C
C *****
C * KBIACC COMPUTES THE BINOMIAL COEFFICIENT, *
C *  $C(N,K) = (N*(N-1)*...*(N-K+1))/(K*(K-1)*...1)$ , *
C * WHERE N AND K ARE THE INTEGER ARGUMENTS TO *
C * THE FUNCTION. INTERMEDIATE CALCULATIONS ARE *
C * PERFORMED IN REAL ARITHMETIC. IN THE CASE *
C * WHERE K .GT. N, A VALUE OF ZERO IS RETURNED. *
C * THE VALUES OF N AND K ARE LEFT UNCHANGED. *
C * THE FUNCTION HAS BEEN CHECKED FOR ALL COM- *
C * BINOMIALS OF N=1,2,...,20 AND K=1,2,...,10. *
C *****
C
ISN 0002      FUNCTION KBIACC (N,K)
C             CHECK FOR TRIVIAL CASES
ISN 0003      IF (K .GT. N) GO TO 50
ISN 0005      IF (K .EQ. 0 .OR. N .EQ. K) GO TO 60
ISN 0007      IF (K .EQ. 1 .OR. N-K .EQ. 1) GO TO 70
C             CONVERT TO REAL FOR INT. CALCULATIONS
ISN 0009      P = N
ISN 0010      Q = K
C             CHECK FOR LOWER 'DENOMINATOR'
ISN 0011      IF (P-Q .LT. 0) Q = P-Q
C             COMPUTE DENOMINATOR
ISN 0013      MAX = 0
ISN 0014      PCT = 1.0
ISN 0015      DO 30 I=2,MAX
ISN 0016      PCT = I*PCT
ISN 0017      30 CONTINUE
C             COMPUTE NUMERATOR
ISN 0018      MAX = P
ISN 0019      MIN = P - Q + 1.0
ISN 0020      TGP = 1.0
ISN 0021      DO 40 I=MIN,MAX
ISN 0022      TGP = I*TGP
ISN 0023      40 CONTINUE
C             CALCULATE AND ROUND BIN. COEFF.
ISN 0024      KBIACC = TGP/PCT + 0.5
ISN 0025      RETURN
ISN 0026      50 KBIACC = 0
ISN 0027      RETURN
ISN 0028      60 KBIACC = 1
ISN 0029      RETURN
ISN 0030      70 KBIACC = N
ISN 0031      RETURN
ISN 0032      END

```

Figure 16. Sample Program Produced by the Library Distribution Tape (Part 2 of 2)

FORTRAN IV (H EXTENDED) COMPILER

The following messages are produced during the installation of the FORTRAN IV (H Extended) compiler. There are two types of messages, information and error. The information messages list the compiler default options that have been set by the execution of the FORTRANX macro instruction. The error messages indicate the incorrect coding of the FORTRANX macro instruction's keyword operands.

INSTALLATION INFORMATION MESSAGES

The messages listed in Table 4 are produced after a successful assembly of the FORTRANX macro instruction. They indicate the keyword operand parameters that have been chosen as system defaults for each compiler option. No programmer response is necessary.

INSTALLATION ERROR MESSAGES

Error messages are produced when a syntax error(s) is encountered during the assembly of the FORTRANX macro instruction. When an error is detected, syntax checking continues; however, the macro is not executed. The error must be corrected and the macro instruction must be reassembled without errors for execution to occur. For a list of the messages, assemble the FORTRANX macro with an operand of IUSTERR=LIST.

Explanation: The message will contain the parameter that was incorrectly coded for the keyword operand of the FORTRAN macro instruction and specify how to code it correctly.

Programmer Response: Correct the syntax of the keyword operand and reassemble. See the section "Coding the FORTRANX Macro Instruction" for more detailed information. If the problem recurs, do the following before calling IBM for programming support:

- Have source deck and associated listing available.

Table 4. Compiler Installation Information Messages (Part 1 of 2)

FORTRANX Macro Operand that Produces Message	Text of Message Produced	Value of xxxxxxx
PUNCH=	PUNCH OPTION FOR OBJECT DECK IS xxxxxxx	DECK or NODECK
SORLIST=	LISTING OPTION FOR SOURCE PROGRAM IS xxxxxxx	SOURCE or NOSOURCE
STORMAP=	RELATIVE LOCATION MAP OPTION IS xxxxxxx	MAP or NOMAP
OBJPROG=	OUTPUT OPTION FOR OBJECT MODULE IS xxxxxxx	OBJECT or NOBJECT
SORCODE=	SOURCE CODE CHARACTER SET OPTION IS xxxxxxx	BCD or EBCDIC
OBJLIST=	LISTING OPTION FOR OBJECT PROGRAM IS xxxxxxx	LIST or NOLIST
OPT=	OBJECT PROGRAM OPTIMIZATION OPTION IS xx OPTIMIZATION	1 NO 2 or BASIC 3 FULL
LINECNT=	OBJECT PROGRAM PRINT PAGE IS MAXIMUM OF xxx LINES	any number from 1 to 200
ADSIZE=	SIZE OF ADCON TABLE IS xxxx BYTES	any number from 1024 to 16384
CMSIZE=	SIZE OF BACKWARD CONNECTOR TABLE IS xxx BYTES	any number from 1024 to 65536
OBJID=	INTERNAL STATEMENT NUMBERING OPTION IS xxxxxxxx	GOSTMT or NOGOSTMT
SOREDIT=	STRUCTURED SOURCE LISTING OPTION IS xxxxxxx	FORMAT or NOFORMAT
SORXREF=	SOURCE CROSS REFERENCE OPTION IS xxxxxxx	XREF or NOXREF
SORALC=	ALIGN COMMON OPTION IS xxxxx	ALC or NOALC
SORANSF=	ANSI ONLY FUNCTION OPTION IS xxxxxx	ANSF or NOANSF
COMOVLY=	COMPILER STRUCTURE SPECIFIED IS xxxxxxx	OVERLAY or NOOVERLAY

Table 4. Compiler Installation Information Messages (Part 2 of 2)

FORTRANX Macro Operand that Produces Message	Text of Messages Produced	Value of xxxxxxx
SORTERM=	TERMINAL MESSAGES OPTION IS xxxxxxxx	TERMINAL or NOTERMINAL
COMPSYS=	COMPILER OPTIMIZED FOR xxxxxxxx SYSTEM	MFT MVT TSO VS1 VS2 or CMS
SORFLAG=	DIAGNOSTIC MESSAGE OPTION IS FLAG(x)	I E or S
COMOPTS=	xxxxxxx OPTIONS CAN BE SELECTED FROM THE LIST: LOAD/NOLOAD EDIT/NOEDIT ID/NOID OPT= LINECNT= SIZE= NAME= TRACE=	NO or OBSOLETE
SIZE=	MAIN STORAGE IS UNRESTRICTED: SIZE (MAX) or MAIN STORAGE IS RESTRICTED: SIZE (xxxxK) BYTES	ANY number from 150 to 9999
NAME=	MAIN PROGRAM NAME OPTION IS xxxxxxxx	name or MAIN
INSTERR=	THE LIST OF ALL INSTALLATION OPTION ERROR MESSAGES WAS REQUESTED: INSTALLATION IS TERMINATED AFTER LIST followed by all error messages or THE LIST OF ALL INSTALLATION OPTION ERROR MESSAGES WAS NOT REQUESTED	LIST or NOLIST
*COMDUMP=	DUMP IS REQUESTED FOR UNRECOVERABLE ERRORS	DUMP or NODUMP
*PERMXL=	EXTENDED LANGUAGE (XL) IS RECOGNIZED	
*TRACE=	TRACE OUTPUT REQUESTED IS TRACE (xxxxxxx)	any number

*This message is printed only if the option is not the default; the message is provided only for IBM personnel responsible for program maintenance.

FORTTRAN IV LIBRARY (MOD II)

The following messages are produced during the installation of the FORTRAN IV Library (Mod II). There are two types of messages, information and error. The information messages list the library default options that have been set by the execution of the FORTLIB macro instruction. The error messages indicate the incorrect coding of the FORTLIB macro instruction's keyword operands.

LIBRARY INFORMATION MESSAGES

The messages listed in Table 5 are produced after a successful assembly of the FORTLIB macro instruction. They indicate the keyword operand parameters that have been chosen as system defaults for each library option. No programmer response is necessary.

Table 5. Library Installation Information Messages

FORTLIB Macro Operand that Produces Message	Text of Message Produced	Meaning of xxxxxxxx
UNTABLE=	I/O UNIT TABLE WILL CONTAIN xx LOGICAL UNITS	any 2-digit number from 08 to 99
OBJERR=	UNIT xx USED FOR 'PRINT' STATEMENTS, ERROR MESSAGES AND DUMPS	any 2-digit number
ONLNRD=	UNIT xx USED FOR 'READ' STATEMENTS	any 2-digit number
ONLNPCH=	UNIT xx USED FOR 'PUNCH' STATEMENTS	any 2-digit number
OPTERR=	OBJECT TIME ERROR HANDLING FACILITY IS xxxxxxxx	EXCLUDE or INCLUDE
ADDNTRY=	xxx ADDITIONAL ERROR ENTRIES IN OPTION TABLE	any 3 digit number from 000 to 598
BOUNDRY=	BOUNDARY ALIGNMENT NOT REQUESTED or BOUNDARY ALIGNMENT SPECIFIED	NOALIGN or ALIGN

LIBRARY ERROR MESSAGES

The error messages listed below are produced when a syntax error(s) is encountered during the assembly of the FORTLIB macro instruction. When an error is detected, syntax checking continues; however, the macro instruction is not executed. The errors must be corrected and the macro instruction must be reassembled without errors for execution to occur.

* * * IEIFOL107 UNTABLE VALUE xx INVALID

Explanation: The field containing xx indicates the parameter that was incorrectly coded for the UNTABLE= keyword operand of the FORTLIB macro instruction.

Programmer Response: Correct the syntax of the UNTABLE= keyword operand and reassemble. See the section "Coding the FORTLIB Macro Instruction" for more detailed information. If the problem recurs, do the following before calling IBM for programming support:

- Have source deck and associated listing available.

* * * IEIFOL109 OBJERR VALUE xx INVALID

Explanation: The field containing xx indicates the parameter that was incorrectly coded for the OBJERR= keyword operand of the FORTLIB macro instruction.

Programmer Response: Correct the syntax of the OBJERR= keyword operand and reassemble. See the section "Coding the FORTLIB Macro Instruction" for more detailed information. If the problem recurs, do the following before calling IBM for programming support:

- Have source deck and associated listing available.

* * * IEIFOL111 ONLNRD VALUE xx INVALID

Explanation: The field containing xx indicates the parameter that was incorrectly coded for the ONLNRD= keyword operand of the FORTLIB macro instruction.

Programmer Response: Correct the syntax of the ONLNRD= keyword operand and reassemble. See the section "Coding the FORTLIB Macro Instruction" for more detailed information. If the problem recurs, do the following before calling IBM for programming support:

- Have source deck and associated listing available.

* * * IEIFOL113 ONLNPCH VALUE xx INVALID

Explanation: The field containing xx indicates the parameter that was incorrectly coded for the ONLNPCH= keyword operand of the FORTLIB macro instruction.

Programmer Response: Correct the syntax of the ONLNPCH= keyword operand and reassemble. See the section "Coding the FORTLIB Macro Instruction" for more detailed information. If the problem recurs, do the following before calling IBM for programming support:

- Have source deck and associated listing available.

* * * IEIF OL114 BOUNDARY VALUE xxxxxxxx INVALID

Explanation: The field containing xxxxxxxx indicates the parameter that was incorrectly coded for the BOUNDARY= keyword operand of the FORTLIB macro instruction.

Programmer Response: Correct the syntax of the BOUNDRY= keyword operand and reassemble. See the section "Coding the FORTLIB Macro Instruction" for more detailed information. If the problem recurs, do the following before calling IBM for programming support:

- Have source deck and associated listing available.

* * * IEIFOL115 OPTERR VALUE xxxxxxxx INVALID

Explanation: The field containing xxxxxxxx indicates the parameter that was incorrectly coded for the OPTERR= keyword operand of the FORTLIB macro instruction.

Programmer Response: Correct the syntax of the OPTERR= keyword operand and reassemble. See the section "Coding the FORTLIB Macro Instruction" for more detailed information. If the problem recurs, do the following before calling IBM for programming support:

- Have source deck and associated listing available.

* * * IEIFOL116 ADDNTRY VALUE xxx INVALID

Explanation: The field containing xxx indicates the parameter that was incorrectly coded for the ADDNTRY= keyword operand of the FORTLIB macro instruction.

Programmer Response: Correct the syntax of the ADDNTRY= keyword operand and reassemble. See the section "Coding the FORTLIB Macro Instruction" for more detailed information. If the problem recurs, do the following before calling IBM for programming support:

- Have source deck and associated listing available.

Where more than one page reference is given, the major reference appears first.

ADDNTRY operand 43,45
 ADSIZE operand 16,19
 assembling modules on release 19 63

BOUNDRY operand 43,44

cataloged procedures
 on SYS1.PROCLIB 13
 storage for 57

CMS installation procedures
 for compiler 27
 for library 47

CMSIZE operand 16,19
 COMDUMP operand 17,24
 COMOPTS operand 16,23
 COMOVLY operand 16,22

compiler
 cataloged procedures for 13
 CSECT sizes 68
 distribution tape under CMS
 format of 9
 steps for running 27
 distribution tape under OS
 format of 9
 effect of 26
 preparation for 12
 steps for running 25
 installation under CMS
 messages 77-82
 procedures 27
 requirements 27
 installation under OS
 messages 77-82
 on private libraries 12-13
 procedures 9
 requirements 9
 on SYS1.LINKLIB 13
 on SYS1.PROCLIB 13
 with SYS1.PPOPTION 13
 user-written installation
 procedure 12
 modules 68
 sample program 56,57
 under CMS 32,73
 under OS 26,72
 storage requirements 56,57
 under CMS 58
 under OS 56,57

COMPSYS operand 16,22
 concatenating data sets (see link library list)

CSECT sizes
 for compiler 68
 for library 70
 using IMBDMAP 68

communications vector table (CVT)
 for Model 85 or 195 62
 modifying 62
 CVT (see communications vector table)

distribution tape
 for compiler under CMS
 format of 9
 steps for running 27
 for compiler under OS
 format of 9
 effect of 26
 preparation for 12
 steps for running 25
 for library under CMS
 format of 37
 steps for running 48
 for library under OS
 format of 37
 effect of 46
 preparation for 40
 steps for running 45

extended error handling facility
 description of 64
 PREFACE for 64
 SETENT for 64
 macro definition for 64,66

FORTLIB macro instruction
 description of 42
 example of 42
 format of 42
 operands for
 ADDNTRY 43,45
 BOUNDRY 43,44
 OBJERR 43,44
 ONLNPC 43,44
 ONINRD 43,44
 OPTERR 43,44
 UNTABLE 43

FORTRANX macro instruction
 description of 14
 example of 14
 format of 16
 operands for
 ADSIZE 16,19
 CMSIZE 16,19
 COMDUMP 17,24
 COMOPTS 16,23
 COMOVLY 16,22
 COMPSYS 16,22
 INSTERR 17,23
 LINECNT 16,19
 NAME 17,23
 OBJID 16,20
 OBJLIST 16,20
 OBJPROG 16,18
 OPT 16,20

PERMXL 17,24
 PUNCH 16,18
 SIZE 17,23
 SORALC 16,21
 SORANSF 16,21
 SORCODE 16,18
 SOREDIT 16,20
 SORFLAG 16,22
 SORLIST 16,18
 SORTERM 16,22
 SORXREF 16,21
 STORMAP 16,18
 TRACE 17,24

IMASPZAP service aid
 for release 19 installation 60
 for Model 85 or 195 installation 62

IMBMDMAP service aid 68
 installation messages
 for compiler 77-79
 for library 80-82
 installation procedures under CMS
 for compiler 27
 for library 47
 installation procedures under OS
 for compiler 9,12
 for library 37,40
 overview of 10,38
 for release 19 59,60
 installation requirements under CMS
 for compiler 27
 for library 47
 installation requirements under OS
 for compiler 9
 for library 37
 for release 19 40
 INSTERR operand 17,23

JOBLIB statement
 for compiler installation 12-13
 for library installation 40,41
 for release 19 installation 61

library
 CSECT sizes 70
 distribution tape under CMS
 format of 37
 steps for running 48
 distribution tape under OS
 format of 37
 effect of 46
 preparation for 40
 steps for running 45
 installation under CMS
 messages 77-82
 procedures 47
 requirements 47
 installation under OS
 messages 77-82
 on private libraries 40,41
 procedures 37
 requirements 37
 on SYS1.FORTLIB 41
 on SYS1.LINKLIB 41

 with SYS1.PPOPTION 41
 user-written installation
 procedure 40
 modules 70
 sample programs 46,74
 storage requirements 57
 LINECNT operand 16,19
 link library list 66

macro definition for extended error
 handling 64
 Model 85 or 195
 installation on 62
 modifying CVT for 62
 modules
 for compiler 68
 for library 70

NAME operand 17,23

OBJERR operand 43,44
 OBJID operand 16,20
 OBJLIST operand 16,20
 OBJPROG operand 16,18
 ONLNPCH operand 43,44
 ONLNRD operand 43,44
 OPT operand 16,20
 OPTERR operand 43,44

PERMXL operand 17,24
 PREFACE for extended error handling 64
 PRIME procedure 28,49
 private libraries
 for compiler installation
 to replace SYS1.LINKLIB 13
 to replace SYS1.PROCLIB 13
 for library installation
 to replace SYS1.FORTLIB 41
 to replace SYS1.LINKLIB 41
 for release 19 installation
 to replace SYS1.LINKLIB 61
 Program Temporary Fix
 for CMS
 compiler 27,30
 library 47,53
 for OS 59,66
 PTF (see program temporary fix)
 PUNCH operand 16,18

release 21
 assembling modules on 63
 extended precision floating point
 simulator 60,62
 installation on 9
 modifying 60
 PTFs for 59,60
 release 20 distribution library 63

sample programs
 for compiler under CMS
 description of 32
 example of 73

for compiler under OS
 description of 26
 example of 72
 for library under CMS 54
 for library under OS
 description of 46
 example of 74
 scratch A disk 28,48
 SETENT for extended error handling 64
 SIZE operand 17,23
 SORALC operand 16,21
 SORANSF operand 16,21
 SORCODE operand 16,18
 SOREDIT operand 16,20
 SORFLAG operand 16,22
 SORLIST operand 16,18
 SORTERM operand 16,22
 SORXREF operand 16,21
 STEPLIB statement
 for compiler 12,13
 for library 27,28
 for release 19 61
 storage requirements
 for compiler under CMS 58
 for compiler under OS
 dynamic 56
 secondary 57
 for library under CMS 58
 for library under OS 57
 for SYS1.FORTLIB 57
 for SYS1.LINKLIB 57
 for SYS1.PPOPTION 57
 for SYS1.PROCLIB 57
 STORMAP operand 16,18
 SYSLIN blocksize 67
 SYS1.FORTLIB
 for library 41
 storage for 57
 SYS1.LINKLIB
 for compiler 13
 for library 41
 for release 19 44
 storage for 57
 SYS1.PPOPTION
 for compiler 13
 for library 41
 storage for 57
 SYS1.PROCLIB
 for compiler 13
 storage for 57

 TRACE operand 17,24

 UNTABLE operand 43
 user-written installation procedure
 for compiler 12
 for library 40
 for release 19 61

READER'S COMMENTS

TITLE: IBM FORTRAN IV
(H Extended) Compiler and
FORTRAN Library (Mod II)
for OS and VM/370 (CMS)
Installation Reference Material

FORM: SC28-6861-2

Your comments assist us in improving the usefulness of our publications; they are an important part of the input used in preparing updates to the publications. All comments and suggestions become the property of IBM.

Please do not use this form for technical questions about the system or for requests for additional publications; this only delays the response. Instead, direct your inquiries or requests to your IBM representative or to the IBM Branch Office serving your locality.

Corrections or clarifications needed:

<i>Page</i>	<i>Comment</i>
-------------	----------------

Please include your name and address in the space below if you wish a reply.

Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

cut along this line

IBM FORTRAN IV (H Ext) Compiler & Mod II Library for OS and VM/370 IRM Printed in U.S.A. SC28-6861-2

fold

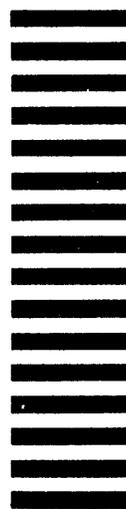
fold

FIRST CLASS
PERMIT NO. 33504
NEW YORK, N.Y.

BUSINESS REPLY MAIL
NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . .

IBM CORPORATION
1271 Avenue of the Americas
New York, New York 10020



Attention: PUBLICATIONS

fold

fold



International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)



International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)