

The PDP-11 FORTRAN-77/IAS Installation Guide explains the installation procedure and contains the release notes for FORTRAN-77/IAS Version 5.0

PDP-11 FORTRAN-77/IAS Installation Guide/Release Notes

**Order No. AA-CJ25A-TC
September 1984**

Supersession/Update Information: This is a new manual for this release.

Operating System and Version: IAS Version 3.2

Software Version: FORTRAN-77 V5.0

**Digital Equipment Corporation
Maynard, Massachusetts**

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

The specifications and drawings, herein, are the property of Digital Equipment Corporation and shall not be reproduced or copied or used in whole or in part as the basis for the manufacture or sale of items without written permission.

Copyright (c) 1984 by Digital Equipment Corporation
All Rights Reserved

The following are trademarks of Digital Equipment Corporation:

CTI BUS	MASSBUS	Rainbow
DEC	PDP	RSTS
DECmate	P/OS	RSX
DECnet	PRO/BASIC	Tool Kit
DECsystem-10	PRO/Communications	UNIBUS
DECSYSTEM-20	Professional	VAX
DECUS	PRO/FMS	VMS
DECwriter	PRO/RMS	VT
DIBOL	PROSE	Work Processor
digital	PROSE PLUS	

CONTENTS

CHAPTER 1	INTRODUCTION	
1.1	SYSTEM REQUIREMENTS	1-1
1.2	DISTRIBUTION FILES	1-2
1.2.1	[1,2]	1-2
1.2.2	[11,36]	1-2
1.2.3	[11,37]	1-2
1.2.4	[11,40]	1-3
CHAPTER 2	INSTALLATION PLANNING	
2.1	SELECTING THE DEFAULT FORTRAN	2-1
2.1.1	Selecting F77 as the Default FORTRAN	2-1
2.1.2	Selecting FOR as the Default FORTRAN	2-2
2.2	SELECTING F77 COMPILER OPTIONS	2-2
2.3	SELECTING F77 OTS OPTIONS	2-2
2.4	PRELIMINARY INSTALLATION PROCEDURES	2-3
CHAPTER 3	SYSTEM TAILORING	
3.1	OPTIONS AFFECTING COMPILE-TIME PERFORMANCE	3-1
3.1.1	Additional Temporary Files	3-1
3.1.2	Selecting the Size of the Dynamic Storage Area	3-2
3.1.2.1	Operating Systems Supporting Dynamic Memory Allocation	3-2
3.2	OTS OPTIONS	3-4
3.2.1	F4P11S	3-4
3.2.2	F4PMAP	3-5
3.2.3	F4PEIS	3-5
3.2.4	F4PCVF	3-6
3.2.5	F4PNER	3-6
3.2.6	F4PNIO	3-6
3.2.7	F4PRAN	3-7
3.2.8	F4PRES	3-7
3.2.9	OTS Overlay Description Files	3-7
CHAPTER 4	INSTALLATION ON IAS USING MCR	
4.1	INSTALLATION FROM MAGNETIC TAPE DISTRIBUTION	4-1
4.1.1	Preparations	4-1
4.1.2	Building the Compiler	4-2
4.1.3	Building the OTS	4-2
4.1.4	Copying the Message Files	4-3
4.2	INSTALLATION FROM DISK CARTRIDGE DISTRIBUTION	4-3
4.2.1	Preparations	4-3
4.2.2	Building the Compiler	4-3
4.2.3	Building the OTS	4-4
4.2.4	Copying the Message Files	4-5
4.2.5	MAKING THE COMPILER AVAILABLE	4-5
CHAPTER 5	INSTALLATION ON IAS USING DCL	
5.1	INSTALLATION FROM MAGNETIC TAPE DISTRIBUTION	5-1
5.1.1	Preparations	5-1
5.1.2	Building the Compiler	5-2
5.1.3	Building the OTS	5-2

5.1.4	Copying the Message Files	5-3
5.2	INSTALLATION FROM DISK CARTRIDGE DISTRIBUTION . .	5-3
5.2.1	Preparations	5-3
5.2.2	Building the Compiler	5-4
5.2.3	Building the OTS	5-4
5.2.4	Copying the Message Files	5-5
5.2.5	MAKING THE COMPILER AVAILABLE	5-5
CHAPTER 6	SYSTEM BUILD VERIFICATION	
6.1	TEST PROGRAM	6-1
6.2	TEST RESULTS	6-2
6.3	VERIFICATION PROCEDURE FAILURE	6-2
CHAPTER 7	OTS RESIDENT LIBRARIES	
7.1	CREATING AN OTS RESIDENT LIBRARY	7-1
7.1.1	General Procedure	7-2
7.1.2	Building a Library with FCS Routines Included .	7-3
7.1.3	Building a Noncluster Library Linked to FCSFSL .	7-3
CHAPTER 8	RELEASE NOTES	
8.1	FORTRAN-77 OTS WITH RMS-11 SUPPORT	8-1
8.2	F77 TYPING OF RESULT-GENERIC FUNCTIONS	8-1
8.3	COMPILER FAILS TO GENERATE WARNING DIAGNOSTIC . .	8-2
8.4	ADJUSTABLE ARRAY USED IN INVALID CONTEXT	8-2
8.5	INTEGER ARRAY REFERENCE USED AS INDEX IN BYTE ARRAY SUBSCRIPT	8-2
8.6	COMMON SUBEXPRESSIONS WITHIN CONDITIONAL STATEMENT	8-3
APPENDIX A	COMPILER TASK-BUILD FILES	
A.1	PDP-11 FORTRAN-77 COMPILER TASK-BUILD FILE FOR IAS (F7711D.CMD)	A-1
INDEX		
FIGURES		
3-1	Compiler Performance	3-3

PREFACE

MANUAL OBJECTIVES

This manual describes the procedures used to install PDP-11 FORTRAN-77 on any PDP-11 system under the IAS operating system, from magnetic tape or disk cartridge distribution kits. PDP-11 FORTRAN-77 can operate on any PDP-11 processor with a Floating-Point Processor (FP11) or floating-point microcode option.

INTENDED AUDIENCE

This manual is intended for the PDP-11 system manager. You should be thoroughly familiar with the IAS operating system before installing the PDP-11 FORTRAN-77 software.

STRUCTURE OF THIS DOCUMENT

This manual is organized as follows:

- o Chapter 1 presents the minimum system requirements for PDP-11 FORTRAN-77 installation on the IAS operating system and lists the software files supplied in the distribution kits.
- o Chapter 2 presents various system options and planning alternatives that must be considered before starting the actual installation process.
- o Chapter 3 explains the effect of selecting certain compiler options described in Chapter 2 and details the optional Object Time System (OTS) modules distributed on the release media.
- o Chapter 4 describes the installation procedures for IAS using MCR command line interpreter.
- o Chapter 5 describes the installation procedures for IAS using DCL command line interpreter.
- o Chapter 6 explains the testing procedure used to verify that your system is correctly installed.
- o Chapter 7 describes OTS resident libraries.
- o Chapter 8 contains release notes.
- o Appendix A presents listings of the compiler task-build files.

PREFACE

ASSOCIATED DOCUMENTS

The PDP-11 FORTRAN-77 User's Guide provides detailed information on using the FORTRAN-77 system. The PDP-11 FORTRAN-77 Language Reference Manual describes the elements of the FORTRAN language as implemented in FORTRAN-77. For information on the Object Time System, consult the PDP-11 FORTRAN-77 Object Time System Reference Manual. For a complete list of software documents, see your host operating-system documentation directory.

CONVENTIONS USED IN THIS MANUAL

The following documentation conventions are used throughout this manual:

\$	ALTMODE	The symbol \$ represents the nonprinting ALTMODE key. This key is called the escape (ESC) key on many terminals. This key is pressed in place of the RETURN key when specified. Unless otherwise specified (that is, with the \$), all commands terminate with a carriage return.
CTRL/X		The notation CTRL/X indicates that you must press the CTRL (control) key while simultaneously typing a letter key (for example, CTRL/C, CTRL/Y, CTRL/O).
UPPERCASE lowercase		In examples, text printed in UPPERCASE characters indicates literal information that must be entered as shown. Text printed in lowercase characters indicates that you are to substitute a word or value of your choice (for example, MTn:).
<RET>		Represents a carriage return.

CHAPTER 1

INTRODUCTION

Section 1.1 of this chapter lists the PDP-11 features and components necessary for successfully building a PDP-11 FORTRAN-77 system. Section 1.2 lists the files contained in the PDP-11 FORTRAN-77 distribution kits.

1.1 SYSTEM REQUIREMENTS

The following sections list the minimum PDP-11 system requirements for installing PDP-11 FORTRAN-77.

NOTE

The term floating-point processor (lowercase) is used in this manual as a generic term to encompass any device or system for processing floating-point data.

The software included in the PDP-11 FORTRAN-77 distribution media requires the following PDP-11 system components for normal use on an IAS operating system:

- o A PDP-11 processor capable of running IAS and equipped with a floating-point processor
- o A minimal 22-K word partition for compilation
- o A minimum of 366 blocks of contiguous on-line disk storage for the compiler task
- o A minimum of 150 to 250 blocks of on-line disk storage for the Object Time System library

Consult Chapter 4 or Chapter 5 for IAS installation instructions.

INTRODUCTION

1.2 DISTRIBUTION FILES

The software necessary for building a PDP-11 FORTRAN-77 system for the IAS operating system is distributed on magnetic tape and disk cartridge media. These media contain the files described in the following sections. The files are listed under the user file directory (UFD) on which they reside in the kit or under their project programmer number (PPN).

1.2.1 [1,2]

File Name	Description
F77COM.MSG	Compiler diagnostic messages file
F77OTS.MSG	Object time system message file

1.2.2 [11,36]

File Name	Description
F77.OLB	PDP-11 FORTRAN-77 compiler object module library
F7711D.CMD	PDP-11 FORTRAN-77 compiler-build command files for
F7711D.ODL	IAS PDP-11 F77 compiler ODL file
F77IAS.CMD	Compiler task-build file

1.2.3 [11,37]

File Name	Description
F4POTS.OBJ	Object Time System concatenated object modules
FCS11D.OBJ	Specific OTS modules for IAS using FCS-11
RMS11D.OBJ	Specific OTS modules for IAS using RMS-11

INTRODUCTION

1.2.4 [11,40]

File Name	Description
F4P11S.OBJ	Specific OTS modules for RSX-11S subset
F4PMAP.OBJ	PDF name-mapping concatenated object module
F4PCVF.OBJ	Optional OTS floating-point formatted output conversion routine
F4PEIS.OBJ	EIS replacements for OTS integer functions that make use of the FPP
F4PNIO.OBJ	Optional OTS modules without FORTRAN I/O capability
F4PNER.OBJ	Optional OTS error-reporting module
F4PRAN.OBJ	Optional random-number generator

OTS Assembly Parameter Files

F77.MAC
FPP.MAC
FPPDP.MAC
FCS.MAC
RMS.MAC
RSXD.MAC

File Name	Description
F77TST.FTN	System installation verification test program
F4PRES.MAC	OTS shared-library source file
FCS11D.ODL	OTS overlay description file for IAS using FCS-11
RMS11D.ODL	OTS overlay description file for IAS using RMS-11

CHAPTER 2

INSTALLATION PLANNING

This chapter discusses some alternatives you must consider before you start the installation process.

2.1 SELECTING THE DEFAULT FORTRAN

If you wish to run both FORTRAN IV (FOR) and PDP-11 FORTRAN-77 (F77) on the same system, you must decide whether FOR or F77 is to be the default FORTRAN. You must make this decision because:

- o When you build a task, object modules produced by the FOR compiler or from the FOR Object Time System (OTS) must not be combined with object modules produced by the F77 compiler or from the F77 OTS.
- o The F77 OTS and the FOR OTS cannot be in the same object module library. (The F77 OTS module is named F4POTS.)

Normally, the default FORTRAN OTS is part of system object module library LB:[1,1]SYSLIB.OLB. The Task Builder searches this library automatically when linking a task. Either the FOR OTS or the F77 OTS can be located in this library.

If you choose to use both FORTRAN systems, you must build a separate library to contain whichever FORTRAN OTS you do not put in LB:[1,1]SYSLIB.OLB. To use this separate library, you must explicitly name it in a Task Builder command line. (The Task Builder always searches the library named in the invoking command line before searching SYSLIB.)

Before selecting a default FORTRAN, consider which of the two FORTRANs you would like to be easier to use. The FORTRAN whose OTS is in SYSLIB will not require an explicit OTS library reference at task-build time, and therefore will consistently save you time and effort.

2.1.1 Selecting F77 as the Default FORTRAN

If you select F77 as the default FORTRAN, you can continue to use the FOR OTS in one of two ways: by building a separate library containing only FOR OTS modules; or by renaming the current SYSLIB, if it contains FOR OTS modules, to LB:[1,1]FOROTS.OLB. Other routines not pertaining to FORTRAN IV may also be in SYSLIB. These routines would have to be referenced as modules in FOROTS.OLB if SYSLIB were renamed.

INSTALLATION PLANNING

2.1.2 Selecting FOR as the Default FORTRAN

If you select FOR as the default FORTRAN, follow the instructions in the IAS/RSX FORTRAN IV Installation Guide for including the FOR OTS in SYSLIB. Then build a separate F77 OTS library, as described in "Building the OTS" in the installation chapter which pertains to your command line interpreter (Chapter 4 or 5).

2.2 SELECTING F77 COMPILER OPTIONS

The following options are available when you build the F77 compiler:

- o You can assign the compiler work file to a device other than the system device, to enhance compiler performance.
- o You can specify the number of pages that are to make up the dynamic storage area to be used by the compiler. A larger paging memory decreases work-file paging activity and enhances compiler performance.
- o You can specify the number of lines per page and the line width for listing devices that will be using non-U.S. standard paper stock.
- o You can specify compiler command switch default settings that differ from the DIGITAL-provided defaults.
- o You can alter the values of the following compiler limits:
 - Number of actual arguments per CALL or function reference
 - Number of named COMMON blocks
 - Number of OPEN statement keywords
 - DO and block IF statement nesting depth
- o You can choose the default setting for the /F77 compiler switch to be either /F77 or /NOF77.

You can edit the compiler-build file to select any of the above options. Documentation within the file describes the options available and any limitations on choices (see Appendix A). Chapter 3 describes the magnitude of the performance improvements that result from various combinations of the first two options.

2.3 SELECTING F77 OTS OPTIONS

The F77 Object Time System supports either of the following file systems for I/O processing:

- o File Control Services (FCS-11), which supports sequential and direct access to sequential files

INSTALLATION PLANNING

- o Record Management Services (RMS-11), which supports sequential, direct, and keyed access to sequential, relative, and indexed files

When you build the OTS, you must choose either the FCS-11 file system or the RMS-11 file system; the F77 OTS does not support both file systems at one time.

The distribution kit contains alternate files for incorporating either FCS-11 or RMS-11.

The files for FCS-11 support are FCS11D.OBJ.

The files for RMS-11 support are RMS11D.OBJ.

Sections 4.1.3 and 4.2.3 include a choice of commands that determines whether your system supports RMS-11 or FCS-11.

Section 3.2 describes OTS options that you can add after you install the basic system.

2.4 PRELIMINARY INSTALLATION PROCEDURES

The following utility programs are required for systems using MCR. If your system uses MCR, install these programs before you proceed with the rest of the installation:

PIP
FLX (if using a magnetic tape distribution)
TKB
LBR

If you are replacing any version of FORTRAN IV-PLUS, the following preliminary steps are required:

1. If the FORTRAN IV-PLUS compiler is installed in the system and you are installing the F77 compiler as F4P, remove it with the following command:

```
REM ...F4P
```

2. If you currently have FORTRAN IV-PLUS, the following files will be on the system disk:

```
[1,2] F4POTS.MSG  
[11,1] F4P.TSK
```

After PDP-11 FORTRAN-77 has been successfully installed, the following files will be on the system disks:

```
[1,2] F77COM.MSG  
[1,2] F77OTS.MSG
```

INSTALLATION PLANNING

3. After PDP-11 FORTRAN-77 has been successfully installed and if you no longer intend to use FORTRAN IV-PLUS, you can delete the following files:

- [1,2] F4PCOM.MSG
- [1,2] F4POTS.MSG
- [11,1] F4P.TSK
- [1,2] F4P.TSK

4. Delete all files in reserved UFD [11,36] where FORTRAN-IV-PLUS resides.
5. If you want to incorporate the F77 OTS into SYSLIB.OLB, obtain a copy of SYSLIB. Save the original SYSLIB.OLB by renaming it to SYSLIB.OLD. The renaming facilitates recovery of the original SYSLIB if the build is unsuccessful.

You must either obtain a copy of SYSLIB that contains everything currently in SYSLIB except the PDP-11 FORTRAN OTS and incorporate the PDP-11 FORTRAN-77 OTS into it, or obtain a fresh copy of SYSLIB from the operating system distribution kit and reincorporate all optional software modules. OTS modules cannot be added to a library containing a previous version of the OTS.

6. If you want to incorporate the F77 OTS into an existing library, make sure that the library module name table and the entry point name table can accommodate the OTS.

- Module names: 200
- Entry points: 450

If necessary, compress the existing library using the librarian compress command.

On an IAS system using MCR or DCL, use a privileged account, for example UIC [1,1], for all operations used in building FORTRAN-77.

CHAPTER 3

SYSTEM TAILORING

This chapter describes options you may choose when building PDP-11 FORTRAN-77 into your system. It includes factors affecting compiler performance and information about optional OTS modules that you can use to tailor PDP-11 FORTRAN-77 to your particular applications.

3.1 OPTIONS AFFECTING COMPILE-TIME PERFORMANCE

There are three options open to you that affect compile-time performance:

- o You may choose one or two additional temporary disk files for the compiler to use for storing information during the compilation process.
- o You may alter the size of the dynamic storage area in the compiler.
- o You may place the compiler work file on a fixed-head disk.

The PDP-11 FORTRAN-77 compiler uses temporary disk files for storing information during the compilation process. The compiler requires at least one temporary file, called the work file.

The work file contains information that the compiler normally accesses at random (for example, the symbol table and the constants table). The dynamic storage area within the compiler is used to manipulate this information. (Only part of the work file is in memory at any given time. Software paging techniques move information back and forth between the dynamic storage area and the work file.)

Information must be moved into the dynamic storage area when needed by the compiler. Therefore, increasing the size of the dynamic storage area increases compilation speed by reducing the number of disk I/O operations (see Section 3.1.3). Similarly, speeding up the average disk I/O operation, by placing the work file on a fixed-head disk instead of a moving-head disk, also improves the compilation rate (see Section 3.1.6).

3.1.1 Additional Temporary Files

The /WF:w compiler switch specifies the number of temporary disk files that are to be available to the compiler. If you specify /WF:1, the compiler stores internal representations of optimized source code and PDP-11 code in just one file, the work file.

SYSTEM TAILORING

However, if you specify /WF:2 (or /WF:3), the compiler stores some (or all) of these representations in the one or two other temporary files. (The /WF:2 option is the default.)

Using additional temporary files slows the compilation process, but it significantly increases the capacity of the compiler. For instance, with three temporary files (/WF:3) available to it, the compiler can compile a program that is approximately three times larger than any it can compile with only one temporary file (/WF:1).

3.1.2 Selecting the Size of the Dynamic Storage Area

Increasing the size of the dynamic storage area increases the rate of compilation. Experimental data indicates that using 14 pages of dynamic storage results in optimal improvement in compilation speed.

Figure 3-1 illustrates the correlation between compile time and the size of the dynamic storage area. The compile time of four different FORTRAN programs, varying in length from 90 to 450 statements, was measured on a PDP-11/60. The compiler used two temporary files (/WF:2), with the work file residing on the system moving-head disk (RP04). The dynamic storage area varied in size from 4 to 26 pages.

No output listing was produced.

The measurements at the end points of each curve denote approximate compilation rate measured in statements compiled per minute. Continuation and comment lines were not counted.

As the figure shows, compilation speed is approximately three times greater when 26 pages of dynamic storage are used than it is when only 4 pages are used. However, using 14 pages results in optimal compiler performance. Building the compiler with more than 14 pages of dynamic storage achieves minimal improvement in the rate of compilation. The default size of the dynamic storage area is 12 pages.

3.1.2.1 Operating Systems Supporting Dynamic Memory Allocation -
Under IAS with dynamic memory allocation, you specify the size of the PDP-11 FORTRAN-77 compiler's dynamic storage area by using the EXTTSK option in the task-build command file. The value specified by EXTTSK is the size of the dynamic storage area in decimal words. The size of the dynamic storage area is computed as follows:

$$256*(n+w+1)$$

n

The number of pages for the dynamic storage area.

w

The value specified in the /WF:w switch.

You can override the dynamic storage area specified by EXTTSK at installation by means of the INC switch on the INSTALL (INS) command. The task extension size is specified in decimal words.

SYSTEM TAILORING

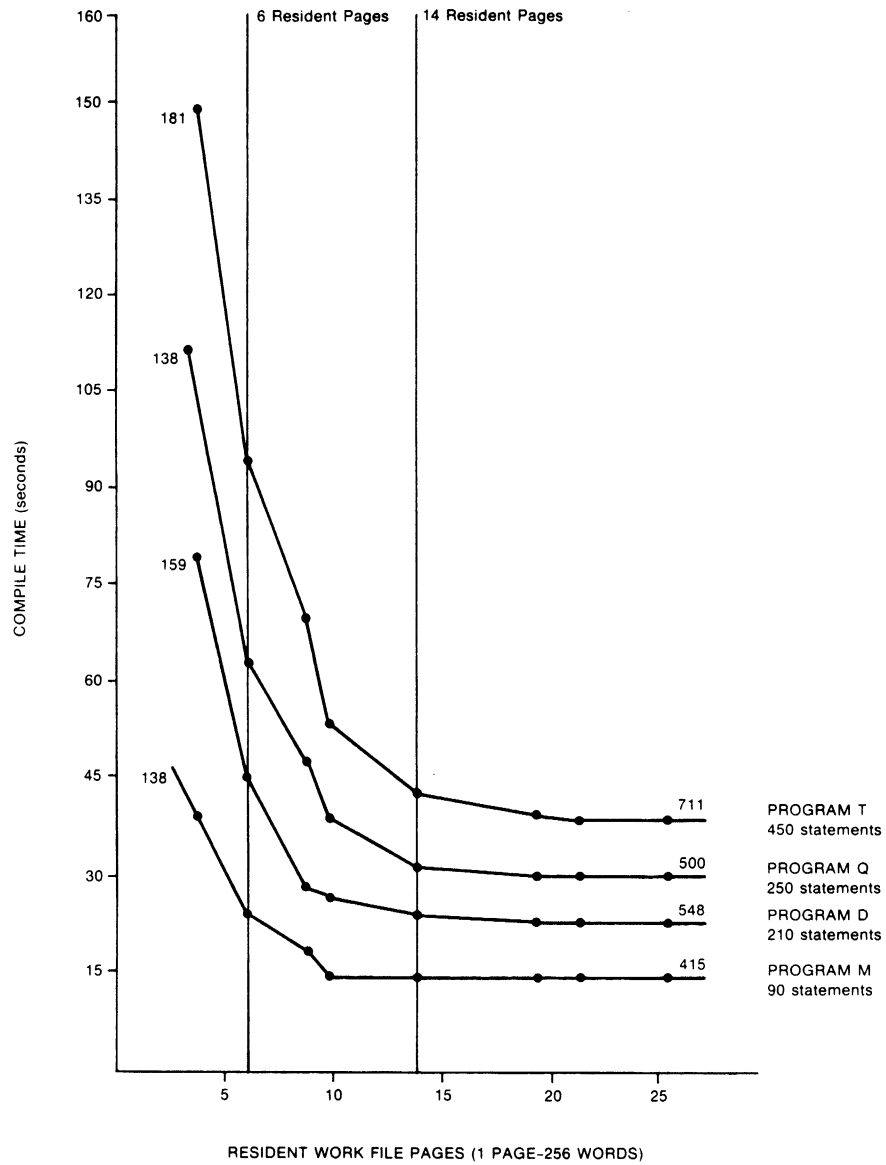


Figure 3-1: Compiler Performance

SYSTEM TAILORING

The following table shows the correlation between the compiler task size, the EXTTSK value, and the number of pages for the dynamic storage area under /WF:2.

Number of Pages	EXTTSK - Value INS/INC Value	Size of Compiler Task Compiler Task (Words)
4	1792	22K
8	2816	23K
12	3840	24K
16	4864	25K

For installations that use by default the ANSI magnetic tape version of FCS-11 (LB:[1,1]ANSLIB.OLB), the compiler task size increases by approximately 500 words.

3.2 OTS OPTIONS

The distribution kit includes a number of optional OTS modules under UFD [11,40]. After building the OTS library, you can add one or more of these optional modules to the library, or you can maintain these modules separately and refer to them only as needed. To add the modules to the library, refer to procedures in "Building the OTS" in either Chapter 4 or 5.

The installation procedures copy these modules to LB:[1,1]. The FORTRAN-77 system does not require any of the optional modules for normal use.

3.2.1 F4P11S

F4P11S.OBJ is a concatenated object module containing alternate versions of FORTRAN sequential I/O support modules. These I/O support modules, designed for use with RSX-11S, provide sequential I/O to non-file-structured devices (for example, terminals, nonspooled card readers, and line printers). These modules do not use the file system but perform direct QIO operations; they reduce task size by approximately 2500 words.

You can use F4P11S.OBJ in two ways:

- o You can include it as an object module at task-build time, as follows:

```
TKB>MAIN/FP=PROG1,LB:[1,1]F4P11S.OBJ
```

- o Or you can build a separate F77 OTS library for RSX-11S use, LB:[1,1]F4P11S.OLB, in addition to the host operating system's OTS library. To do this, when building the OTS, substitute module LB:[1,1]F4P11S.OBJ for the file system module selected in Section 2.3. For example, replace the reference to FCS11M.OBJ with LB:[1,1]F4P11S.OBJ.

Use this OTS library, rather than the host operating system's OTS library, when building tasks for RSX-11S, as follows:

```
TKB>PROG1/FP=PROG1,LB:[1,1]F4P11S/LB
```

SYSTEM TAILORING

3.2.2 F4PMAP

Module F4PMAP.OBJ consists of a set of concatenated object modules that can be used to transform intrinsic function names into internal names at task-build time. (The PDP-11 FORTRAN-77 compiler transforms intrinsic function names into internal names at compile time.)

Without F4PMAP.OBJ, if a program written in MACRO-11 attempts to reference a PDP-11 FORTRAN-77 intrinsic function with the FORTRAN name of the function instead of the internal name, an unresolved reference will occur during task build.

For example, F4PMAP.OBJ maps the FORTRAN name SIN by means of the following module:

```
SIN::      .TITLE      $MSIN
           JMP          $SIN
           .END
```

F4PMAP.OBJ contains an object module similar to the above module for each of the PDP-11 FORTRAN-77 intrinsic functions.

You can build an F4PMAP library as follows:

```
(MCR command format)
LBR>LB:[1,1]F4PMAP.OLB/CR:40.=LB:[1,1]F4PMAP.OBJ

(IAS PDS command format)
PDS>LIBR CREATE/SIZE:40 LB:[1,1]F4PMAP LB:[1,1]F4PMAP
```

3.2.3 F4PEIS

F4PEIS.OBJ is a concatenated object module containing extended instruction set (EIS) versions of certain integer functions that normally use a floating-point processor. This module allows FORTRAN programs that do not do floating-point arithmetic to run on a machine that has the extended instruction set but not a floating-point processor. The modules provided in the F77 OTS use a floating-point processor for maximum efficiency in certain INTEGER*4 computations.

Use one of the following commands to replace at task-build time the normal modules in file INTEGER with their EIS versions:

```
(MCR command format)
TKB>INTEGER/-FP=INTEGER, LB:[1,1]F4PEIS.OBJ
```

SYSTEM TAILORING

3.2.4 F4PCVF

Object module F4PCVF.OBJ is an alternative module for performing formatted output of floating-point values under control of the D, E, F, and G field specifiers. The standard module provided as part of the F77 OTS uses multiple-precision, fixed-point integer techniques to maintain maximum accuracy during the conversion of data (FPP hardware is not used). The alternative module performs the same functions using the FPP hardware. It is approximately twice as fast as, but in some cases slightly less accurate than, the standard module.

You can substitute the F4PCVF module for the default conversion module as follows:

```
(MCR command format)
LBR>LB:[1,1]SYSLIB=LB:[1,1]F4PCVF/RP
```

3.2.5 F4PNER

Object module F4PNER.OBJ is an alternative module for reporting run-time errors. If you use this module, the error-message text report is suppressed. However, error processing and calls to ERRSET, ERRSNS, and ERRST continue to operate normally; only the logging of the message on the user's terminal is suppressed. The STOP and PAUSE statement messages are also suppressed. F4PNER.OBJ reduces task size by about 375 words over the standard module.

If you use F4PNER with F4P11S or F4PNIO -- other optional OTS modules -- a multiply defined symbol error may result during task-build. Two correct ways to use F4PNER with F4P11S or F4PNIO follow:

- o Build F4P11S (or F4PNIO) and F4PNER as separate libraries and use them as follows:

```
TKB>PROG1/FP=PROG1, LB:[1,1]F4PNER/LB:$NERRL,
      LB:[1,1]F4P11S/LB, LB:[1,1]F4POTS/LB
```

- o Build an OTS by incorporating F4P11S.OBJ instead of FCS11M.OBJ into F4POTS. Name the resulting library F4P11S.OLB, and build F4PNER as a separate library. Use those libraries as follows:

```
TKB>PROG1/FP=PROG1, LB:[1,1]F4PNER/LB:$NERRL,
      LB:[1,1]F4P11S/LB
```

3.2.6 F4PNIO

F4PNIO.OBJ is a concatenated object module containing alternative versions of certain OTS routines that are always present in the user task and that provide support for FORTRAN I/O operations. The alternate routines in F4PNIO.OBJ do not support FORTRAN I/O and reduce task size by approximately 1000 words for programs that do not require FORTRAN I/O (such as process control).

SYSTEM TAILORING

3.2.7 F4PRAN

F4PRAN.OBJ is a concatenated object module containing an alternative random-number generator that is compatible with previous releases of PDP-11 FORTRAN. If you require this random-number generator for compatibility purposes, include file LB:[1,1]F4PRAN.OBJ at task-build time.

3.2.8 F4PRES

F4PRES.MAC is a MACRO-11 source file containing global references to all modules of the OTS. An OTS resident library (shared global area, or SGA) is an option available to installations that have many FORTRAN-77 programs as well as PDP-11 FORTRAN-77 development. F4PRES.MAC contains the global references to the OTS and documentation on logical groups of OTS modules. This feature aids in the building of an OTS resident library that is tailored to the needs of a particular installation.

See Chapter 7 for more information about building and using OTS resident libraries.

3.2.9 OTS Overlay Description Files

The two OTS overlay description files are:

- FCS11D.ODL - FCS-11 support for IAS
- RMS11D.ODL - RMS-11(K) support for IAS

Each file is an ODL fragment file that you can use for overlaying the PDP-11 FORTRAN-77 OTS modules. Each file contains documentation that describes OTS options as well as procedures for using the file.

CHAPTER 4

INSTALLATION ON IAS USING MCR

This chapter describes the procedures for installing FORTRAN-77 on an IAS operating system using the MCR command language. Read the sections relevant to the type of distribution kit you are using (Section 6.1 for the magnetic-tape distribution; Section 6.2 for the disk-cartridge distribution). Section 6.3 gives instructions for invoking the compiler.

The basic installation procedure for PDP-11 FORTRAN-77 consists of:

1. Building the F77 compiler task from an object-module library
2. Building an F77 OTS library from object modules
3. Copying the diagnostic message files from the distribution medium to the system disk

4.1 INSTALLATION FROM MAGNETIC TAPE DISTRIBUTION

The following sections explain how to build PDP-11 FORTRAN-77 from a magnetic-tape distribution kit.

4.1.1 Preparations

UFD [11,36] is used in building the PDP-11 FORTRAN-77 system; therefore, if this user file directory is not already present on your system, create it on the system device as follows:

```
MCR>UFD SY:[11,36]
```

Now place the distribution magnetic tape, write-locked, on drive 0. and load the magnetic tape handler, if not already resident, as follows (Note that the device mnemonic for some magnetic tape units is MM:. If you are using such a device, substitute MM: for MT: in the following commands.):

```
MCR>LOA MT:
```

Finally, mount the magnetic tape as follows:

```
MCR>MOU MT0:/CHA=[FOR,ATCH]
```

INSTALLATION ON IAS USING MCR

4.1.2 Building the Compiler

You build the F77 compiler from the object-module library supplied on the distribution medium. First, copy the required files from the magnetic tape to the system disk as follows:

```
MCR>FLX SY:[11,36]=MT0:[11,36]F77.OLB
MCR>FLX SY:[11,36]=MT0:[11,36]F7711D.*
```

Then edit compiler task-build command file [11,36]F7711D.CMD to select installation options, as described in Section 2.2 and in Appendix A.

Now build the compiler as follows:

```
MCR>TKB @[11,36]F7711D
```

Retain the edited command files in [11,36] for reference.

4.1.3 Building the OTS

To build the OTS, first copy the required OTS files from the magnetic tape to the system disk as follows:

```
MCR>FLX SY:[11,36]=MT0:[11,37]*.OBJ
```

Now you can build the F77 OTS in either of two ways:

1. One way to build the OTS is to merge the OTS concatenated object modules into an existing library (for example, LB:[1,1]SYSLIB.OLB). If you plan to use the RMS-11 file system and RMSLIB.OLB has already been incorporated into SYSLIB.OLB, you cannot place F4POTS.OLB in SYSLIB.OLB.

First, use the LBR utility program as follows:

```
MCR>LBR LB:[1,1]SYSLIB.OLB/RP=[11,36]F4POTS.OBJ
```

Now use one of the following commands to incorporate either RMS-11 support or FCS-11 support, but not both (see Section 2.3):

```
MCR>LBR LB:[1,1]SYSLIB.OLB/RP=[11,36]FCS11D.OBJ
```

or

```
MCR>LBR LB:[1,1]SYSLIB.OLB/RP=[11,36]RMS11D.OBJ
```

2. The second way to build the OTS is to generate a separate library using the OTS concatenated object modules, as follows (If you build a separate library, you must designate file LB:[1,1]F4POTS.OLB when building a task from modules produced by the PDP-11 FORTRAN-77 compiler.):

```
MCR>LBR LB:[1,1]F4POTS.OLB/CR:150.= [11,36]F4POTS.OBJ
```

Now use one of the following commands to incorporate either RMS-11 support or FCS-11 support, but not both:

```
MCR>LBR LB:[1,1]F4POTS.OLB=[11,36]FCS11D.OBJ
```

or

```
MCR>LBR LB:[1,1]F4POTS.OLB=[11,36]RMS11D.OBJ
```

After completing one of the above, copy the optional OTS modules to the system disk as follows:

```
MCR>FLX LB:[1,1]=MT0:[11,40]*.OBJ,*.ODL,*.MAC,*.FTN
```


INSTALLATION ON IAS "USING MCR

4.1.4 Copying the Message Files

At this point, copy the FORTRAN-77 compiler message files to the system disk as follows:

```
MCR>FLX LB:[1,2]=MT0:[1,2]F77COM.MSG/IM:64./BL:14.  
MCR>FLX LB:[1,2]=MT0:[1,2]F77OTS.MSG/IM:64./BL:16.
```

4.2 INSTALLATION FROM DISK CARTRIDGE DISTRIBUTION

The following sections explain how to build PDP-11 FORTRAN-77 from a disk-cartridge distribution kit. If the system disk is not the same device type as that of the distribution disk, place the distribution disk in drive 0. If the system disk is the same device type as that of the distribution disk, use drive 0 for the system disk and drive 1 for the distribution disk. Procedures for these two configurations are the same, except for the different unit assignments.

In the directions that follow, the notation "Dxn" refers to the disk drive unit on which the PDP-11 FORTRAN-77 distribution pack is placed. For this notation, x can take the value K, M, or L, and n can take the value 0 or 1 (on the basis of the system device type). Use DK0 or DK1 for the RK05 distribution; use DM0 or DM1 for the RK06 distribution; and use DL0 or DL1 for the RL01 OR RL02 distribution.

4.2.1 Preparations

UFD [11,36] is used in building the PDP-11 FORTRAN-77 system; therefore, if this user file directory is not already present on your system, create it on the system device as follows, using the following command:

```
MCR>UFD SY:[11,36]
```

Now load the disk handler, if not already resident, as follows:

```
MCR>LOA Dxn:      (not needed if the system device is the same  
                  device type as that of the distribution disk)
```

Finally, place the distribution disk cartridge in drive n, write-locked, and mount the volume as follows:

```
MCR>MOU Dxn:F77
```

4.2.2 Building the Compiler

You build the compiler from the object-module library supplied on the distribution disk. First, copy the compiler build files from the distribution disk to the system disk as follows:

```
MCR>PIP SY:[11,36]=Dxn:[11,36]F7711D.*  
MCR>PIP SY:[11,36]=Dxn:[11,36]F77.OLB
```

Now edit compiler task-build command file [11,36]F7711D.CMD to select installation options, as described in Section 2.2 and in Appendix A.

INSTALLATION ON IAS USING MCR

Edit the compiler overlay-description file [11,36]F7711D.ODL to specify the disk unit on which the distribution disk is mounted, as follows:

```
MCR>EDI SY:[11,36]F7711D.ODL
*PA/SY:/Dxn:/
*EX
```

Now build the compiler as follows:

```
MCR>TKB @[11,36]F7711D
```

Retain the edited command files in [11,36] for reference.

4.2.3 Building the OTS

You can build the F77 OTS in either of two ways:

1. One way to build the OTS is to merge the OTS concatenated object file into an existing library (for example, LB:[1,1]SYSLIB.OLB). If you plan to use the RMS-11 file system and RMSLIB.OLB has already been incorporated into SYSLIB.OLB, you cannot place F4POTS.OLB in SYSLIB.OLB.

First, use the LBR utility program as follows:

```
MCR>LBR LB:[1,1]SYSLIB.OLB/RP=Dxn:[11,37]F4POTS.OBJ
```

Now use one of the following commands to incorporate either RMS-11 support or FCS-11, but not both (see Section 2.3):

```
MCR>LBR LB:[1,1]SYSLIB.OLB/RP=Dxn:[11,37]FCS11D.OBJ
```

or

```
MCR>LBR LB:[1,1]SYSLIB.OLB/RP=Dxn:[11,37]RMS11D.OBJ
```

2. The second way to build the OTS is to generate a separate library using the OTS concatenated object modules, as follows: (If you build a separate library, you must designate file LB:[1,1]F4POTS.OLB when building a task from modules produced by the PDP-11 FORTRAN-77 compiler.)

```
MCR>LBR LB:[1,1]F4POTS.OLB/CR:150.=Dxn:[11,37]F4POTS.OBJ
```

Now use one of the following commands to incorporate either RMS-11 support or FCS-11 support, but not both:

```
MCR>LBR LB:[1,1]F4POTS.OLB=Dxn:[11,37]FCS11D.OBJ
```

or

```
MCR>LBR LB:[1,1]F4POTS.OLB=Dxn:[11,37]RMS11D.OBJ
```

After completing one of the above, copy the optional OTS modules to the system disk as follows:

```
MCR>PIP LB:[1,1]=Dxn:[11,40]*.OBJ,*.ODL,*.MAC,*.FTN
```

INSTALLATION ON IAS USING MCR

4.2.4 Copying the Message Files

At this point, copy the FORTRAN-77 compiler message files to the system disk as follows:

```
MCR>PIP LB:[1,2]=Dxn:[1,2]F77COM.MSG,F77OTS.MSG
```

4.2.5 MAKING THE COMPILER AVAILABLE

After the compiler task has been created, the task-image file must be made available for use (installed). The following section tells how to install the FORTRAN-77 compiler.

On an IAS system using MCR, you install the F77 compiler as follows:

```
MCR> [11,1]F77
```

Refer to the IAS System Generation and Start-up Guide for procedures to follow to save the system image with the F77 compiler installed.

To install PDP-11 FORTRAN-77 as F4P, which may be desirable if your site has existing command files that call the F4P, use the following command:

```
MCR> INS [11,1]F77/TASK=...F4P
```

Invoke the FORTRAN-77 compiler as an MCR function, as follows:

```
MCR>F77
```


CHAPTER 5

INSTALLATION ON IAS USING DCL

This chapter describes the procedures for installing FORTRAN-77 on an IAS operating system using the DCL command language. Read the sections relevant to the type of distribution kit you are using (Section 7.1 for the magnetic-tape distribution; Section 7.2 for the disk-cartridge distribution). Section 7.3 gives instructions for invoking the compiler.

The basic installation procedure for PDP-11 FORTRAN-77 consists of:

1. Building the F77 compiler task from an object-module library
2. Building an F77 OTS library from object modules
3. Copying the diagnostic message files from the distribution medium to the system disk

5.1 INSTALLATION FROM MAGNETIC TAPE DISTRIBUTION

The following sections explain how to build PDP-11 FORTRAN-77 from a magnetic-tape distribution kit.

5.1.1 Preparations

UFD [11,36] is used in building the FORTRAN-77 system; therefore, if this user file directory is not already present on your system, create it on the system device as follows

```
PDS>CREATE/DIR [11,36]
```

Now place the distribution magnetic tape, write-locked, on drive 0 and load the magnetic tape handler, if not already resident, as follows (Note that the device mnemonic for some magnetic tape units is MM:. If you are using such a device, substitute MM: for MT: in the following commands.):

```
SCI>RUN/HANDLER MT
```

Finally, mount the magnetic tape as follows:

```
PDS>MOUNT/FOREIGN MT0: F77
```

INSTALLATION ON IAS USING DCL

5.1.2 Building the Compiler

You build the F77 compiler from the object-module library supplied on the distribution medium. First, copy the needed files from the magnetic tape to the system disk as follows:

```
PDS>COPY MT0:[11,36]F77.OLB/DO [11,36]*.*
PDS>COPY MT0:[11,36]F77IAS.* /DO [11,36]*.*
```

Now edit compiler task-build command file [11,36]F77IAS.CMD to select installation options, as described in Section 2.2 and in Appendix A.

Then build the compiler as follows:

```
PDS>LINK @[11,36]F77IAS
```

Retain the edited command files in [11,36] for reference.

5.1.3 Building the OTS

To build the OTS, first copy the required OTS files from the magnetic tape to the system disk as follows:

```
PDS>COPY MT0:[11,37]*.OBJ/DO [11,36]*.*
```

You can now build the F77 OTS in either of two ways:

1. One way to build the OTS is to merge the OTS concatenated object modules into an existing library (for example, LB:[1,1]SYSLIB.OLB). If you plan to use the RMS-11 file system and RMSLIB.OLB has already been incorporated into SYSLIB.OLB, you cannot place F4POTS.OLB in SYSLIB.OLB.

First, use the LIBR utility program as follows:

```
PDS>LIBR REPLACE LB:[1,1]SYSLIB [11,36]F4POTS
```

Now use one of the following commands to incorporate either RMS-11 support or FCS-11 support, but not both (see Section 2.3):

```
PDS>LIBR REPLACE LB:[1,1]SYSLIB [11,36]FCSIAS.OBJ
```

or

```
PDS>LIBR REPLACE LB:[1,1]SYSLIB [11,36]RMSIAS.OBJ
```

2. The second way to build the OTS is to generate a separate library using the OTS concatenated object modules, as follows (If you build a separate library, you must designate file LB:[1,1]F4POTS.OLB when building a task from modules produced by the PDP-11 FORTRAN-77 compiler.):

```
PDS>LIBR CREATE/SIZE:150 LB:[1,1]F4POTS [11,36]F4POTS
```

Now use one of the following commands to incorporate either RMS-11 support or FCS-11 support, but not both:

```
PDS>LIBR INSERT LB:[1,1]F4POTS [11,36]FCSIAS.OBJ
```

or

```
PDS>LIBR INSERT LB:[1,1]F4POTS [11,36]RMSIAS.OBJ
```

INSTALLATION ON IAS USING DCL

After completing one of the above, copy the optional OTS modules to the system disk as follows:

```
PDS>COPY MT0:[11,40]*.OBJ/DO LB:[1,1]*.*
PDS>COPY MT0:[11,40]*.ODL/DO LB:[1,1]*.*
PDS>COPY MT0:[11,40]*.MAC/DO LB:[1,1]*.*
PDS>COPY MT0:[11,40]*.FTN/DO LB:[1,1]*.*
```

5.1.4 Copying the Message Files

At this point, copy the PDP-11 FORTRAN-77 compiler message files to the system disk as follows:

```
PDS>COPY/IMAGE:64./BLOCKSIZE:14 MT0:[1,2]F77COM.MSG/DOS LB:[1,2]*.*
PDS>COPY/IMAGE:64./BLOCKSIZE:16 MT0:[1,2]F77OTS.MSG/DOS LB:[1,2]*.*
```

5.2 INSTALLATION FROM DISK CARTRIDGE DISTRIBUTION

The following sections explain how to build FORTRAN-77 from a disk-cartridge distribution kit. If the system disk is not the same device type as that of the distribution disk, place the distribution disk in drive 0. If the system disk is the same device type as that of the distribution disk, use drive 0 for the system disk and drive 1 for the distribution disk. Procedures for the two configurations are the same, except for the different unit assignments.

In the directions that follow, the notation "Dxn" refers to the disk drive unit on which the FORTRAN-77 distribution disk is placed. In this notation, x can take the value K, M, or L, and n can take the value 0 or 1. Use DK0 or DK1 for the RK05 distribution; use DM0 or DM1 for the RK06 distribution; and use DL0 or DL1 for the RL01 or RL02 distribution.

5.2.1 Preparations

UFD [11,36] is used in building the PDP-11 FORTRAN-77 system; therefore, if this user file directory is not already present on your system, create it on the system device as follows:

```
PDS>CREATE/DIR [11,36]
```

Now load the disk handler, if not already resident, as follows:

```
SCI>RUN/HANDLER Dx:
```

Finally, place the distribution disk cartridge in drive n (write-locked), and mount the volume as follows:

```
PDS>MOUNT Dxn: F77
```

INSTALLATION ON IAS USING DCL

5.2.2 Building the Compiler

You build the compiler from the object-module library supplied on the distribution disk. First, copy the compiler build files from the distribution disk to the system disk as follows:

```
PDS>COPY Dxn:[11,36]F77IAS.* [11,36]*.*
PDS>COPY Dxn:[11,36]F77.OLB [11,36]*.*
```

Now edit compiler task-build command file [11,36]F77IAS.CMD to select installation options, as described in Section 2.2 and in Appendix A.

Edit the compiler overlay-description file [11,36]F77IAS.ODL to specify the disk unit on which the distribution disk is mounted, as follows:

```
PDS>EDIT [11,36]F77IAS.ODL
*PA/SY:/Dxn:/
*EX
```

Now build the compiler as follows:

```
PDS>LINK @[11,36]F77IAS
```

Retain the edited command files in [11,36] for reference.

5.2.3 Building the OTS

You can build the F77 OTS in either of two ways:

1. One way to build the OTS is to merge the OTS concatenated object modules into an existing library (for example, LB:[1,1]SYSLIB.OLB). If you plan to use the RMS-11 file system and RMSLIB.OLB has already been incorporated into SYSLIB.OLB, you cannot place F4POTS.OLB in SYSLIB.OLB.

First, use the LIBR utility program as follows:

```
PDS>LIBR REPLACE LB:[1,1]SYSLIB Dxn:[11,37]F4POTS
```

Now use one of the following commands to incorporate either RMS-11 support or FCS-11 support, but not both (see Section 2.3):

```
PDS>LIBR REPLACE LB:[1,1]SYSLIB Dxn:[11,37]FCSIAS.OBJ
or
PDS>LIBR REPLACE LB:[1,1]SYSLIB Dxn:[11,37]RMSIAS.OBJ
```

2. The second way to build the OTS is to generate a separate library using the OTS concatenated object modules, as follows (If you use a separate library, you must designate file LB:[1,1]F4POTS.OLB when building a task from modules produced by the PDP-11 FORTRAN-77 compiler.)

```
PDS>LIBR CREATE/SIZE:150 LB:[1,1]F4POTS Dxn:[11,37]F4POTS
```

Now use one of the following commands to incorporate either RMS-11 support or FCS-11 support, but not both:

```
PDS>LIBR INSERT LB:[1,1]F4POTS Dxn:[11,37]FCSIAS.OBJ
or
PDS>LIBR INSERT LB:[1,1]F4POTS Dxn:[11,37]RMSIAS.OBJ
```


INSTALLATION ON IAS USING DCL

After completing one of the above, copy the optional OTS modules to the system disk as follows:

```
PDS>COPY Dxn:[11,40] .OBJ LB:[1,1]*.*
PDS>COPY Dxn:[11,40]*.ODL LB:[1,1]*.*
PDS>COPY Dxn:[11,40]*.MAC LB:[1,1]*.*
PDS>COPY Dxn:[11,40]*.FTN LB:[1,1]*.*
```

5.2.4 Copying the Message Files

At this point, copy the PDP-11 FORTRAN-77 compiler and OTS message files to the system disk as follows:

```
PDS>COPY Dxn:[1,2]*.MSG LB:[1,2]*.*
```

5.2.5 MAKING THE COMPILER AVAILABLE

After the compiler task has been created, the task-image file must be made available for use (installed). The following section tells how to install the FORTRAN-77 compiler.

On an IAS system using DCL, you install the F77 compiler as follows:

```
PDS>INSTALL [11,1]F77
```

If the F77 compiler is installed as F77, you can invoke it only with MCR; the DCL FORTRAN/F77 command is not supported.

To install PDP-11 FORTRAN-77 as the default FORTRAN, and name it FOR instead of F77, use the following command:

```
PDS>INSTALL/TASK:...FOR [11,1]F77
```

To install PDD-11 FORTRAN-77 as F4P, which may be desirable if your site has existing command files that call the F4P compiler, use the followiin command:

```
PDS> INSTALL/TASK:...F4P [11,1]F77
```

Refer to the IAS System Generation and Start-up Guide for procedures required to save the system image with the F77 compiler installed.

You invoke the PDP-11 FORTRAN-77 compiler using the PDS FORTRAN command, if the system default is PDP-11 FORTRAN-77, as follows:

```
PDS>FORTRAN
```

If the compiler is installed as F4P, invoke it as follows:

```
PDS>FORTRAN/F4P
```

Otherwise, you must use MCR to invoke the PDP-11 FORTRAN-77 compiler, as follows:

```
PDS> MCR F77
```


CHAPTER 6

SYSTEM BUILD VERIFICATION

The PDP-11 FORTRAN-77 distribution kit includes a simple test program to verify that the system is correctly installed and ready for use. The test program, F77TST.FTN, contains two intentional nonfatal errors, one occurring during compilation and one during execution. The two errors are documented in Section 6.2.

6.1 TEST PROGRAM

The test program is copied from the distribution medium to the system disk when you build the OTS (see section "Building the OTS in Chapter 4 or 5, depending upon the command language interpreter used). You need to compile, link, and execute F77TST.FTN as described in this section.

MCR Commands

The following command sequences show you how to compile, task-build, and execute the test program on IAS systems.

Compile: F77 F77TST=F77TST/TR:ALL

Link: if using an OTS located in SYSLIB,

(if using FCS-11)

TKB F77TST/FP=F77TST

(if using RMS-11)

TKB F77TST/FP=F77TST,LB:[1,1]SYSLIB.OLB/LIB,LB:[1,1]RMSLIB/LB

or, if a separate library is used,

(if using FCS-11)

TKB F77TST/FP=F77TST,LB:[1,1]F4POTS/LB

(if using RMS-11)

TKB F77TST/FP=F77TST,LB:[1,1]F4POTS/LIB,LB:[1,1]RMSLIB/LB

Execute: RUN F77TST

SYSTEM BUILD VERIFICATION

6.2 TEST RESULTS

During compilation, an overflow error should occur on source line 6. During execution, a floating zero-divide error should be reported at line 8.

During execution on a non-FPP machine, the following run-time error is printed:

```
-- Exiting due to ERROR 7
Reserved instruction trap (SST4)
at PC=xxxxxx
```

6.3 VERIFICATION PROCEDURE FAILURE

The preceding verification procedure can fail if the FORTRAN-77 system is incorrectly installed. Possible failures include:

- o The F77 compiler produces the message:

```
F77 -- FATAL 08 * COMPILER DYNAMIC MEMORY OVERFLOW
```

This message indicates that dynamic memory for the compiler was not correctly allocated by EXTTSK or INS/INC (see Section 3.1.4). Remove and reinstall the compiler with a correct INC value.

- o If the compiler message file, LB:[1,2]F77COM.MSG has not been transferred correctly, the compiler treats the compile-time overflow in F77TST as a fatal error. The compiler does not create object module F77TST.OBJ.
- o If the OTS message file LB:[1,2]F77OTS.MSG has not been transferred correctly or if the MO message task has not been loaded, run time subroutine traceback information is not produced.

CHAPTER 7

OTS RESIDENT LIBRARIES

This chapter describes how you can create and use OTS resident libraries. In particular, it discusses reasons for using (and limitations of) various types of resident libraries, including supervisor-mode libraries.

A resident library has the following characteristics:

- o It resides in memory. It must be installed before a task that references it can be installed or run.
- o It can be shared by multiple tasks. However, it occupies virtual address space in each task which is linked to it.

The FORTRAN-77 OTS has the following general limitations:

- o It does not contain position-independent code (PIC) and therefore cannot be built into a PIC resident library.

For more information on resident libraries, refer to the IAS Task Builder Reference Manual.

7.1 CREATING AN OTS RESIDENT LIBRARY

This section presents the OTS resident library. Section 7.1.1 outlines the general procedure that you follow to build any OTS resident library. Section 7.1.2 provides specific instructions for the various libraries. These sections include Task Builder command files that will build appropriate libraries in most situations.

NOTE

The command files listed in this chapter do not link either of the two OTS error message modules, \$ERTXT or \$SHORT, into the resident library. You may include one of these modules when building your library if you wish to force long or short error message text to be used by programs that link to that library.

OTS RESIDENT LIBRARIES

7.1.1 General Procedure

The following list contains the steps you follow to create an OTS resident library. The text that follows the list contains elaborations on some of the steps. Before you start performing these steps, you should have read Section 7.1 and decided what kind of resident library you need.

1. Edit the supplied file F4PRES.MAC, if necessary, to customize the contents of the OTS resident library to suit your needs. Further information on F4PRES.MAC appears following this list.
2. Assemble F4PRES.MAC, giving F4PRES.OBJ as the output file.
3. Use a text editor to create the appropriate Task Builder command file to build your resident library. (The command files in Sections 7.2.2 through 7.2.5 should work either as is or with slight modifications.) Make sure that the OTS object module library that you specify in the command file contains file system modules (either FCS or RMS) that match the file system you intend to use.
4. Invoke the Task Builder and pass it the command file you just created.
5. Inspect the map file resulting from the task build. If the resident library is too large or is not large enough, edit (or reedit) F4PRES.MAC and repeat the steps outlined above. (More information on recommended library size appears following this list.)
6. Purge any task, map, and STB files resulting from previous task builds.
7. Install the library in your system, following the instructions in the documentation for your particular operating system.

The file F4PRES.MAC (which is available on the FORTRAN-77 kit) contains global references to OTS entry points. The modules referenced in F4PRES.MAC will make up the OTS resident library. You can edit F4PRES.MAC to include modules that your tasks use frequently, or to exclude modules that are used infrequently. (Editing instructions are included in the file.)

If you edit F4PRES.MAC, your goal should be to create an OTS resident library that appropriately balances the requirements of size and functionality. If your library is very large, the virtual address space available for your task may be unreasonably small. (A maximum library size of 8K words is recommended.) On the other hand, the Task Builder will place in your task the object code for any modules it references that are not in the OTS resident library; thus, it does not make sense to exclude commonly used modules from the library.

To make the best use of available virtual memory, the OTS resident library should be nearly equal to but slightly below a multiple of 4K words. Each time the size of the library exceeds a 4K multiple, an additional APR is required; this has the effect of reducing the virtual address space available to the task by an entire 4K words. The following table illustrates this relationship:

OTS RESIDENT LIBRARIES

Size in Words	Number of APRs	Size in Octal Bytes
4096	1	20000
8192	2	40000
12288	3	60000
16384	4	100000

7.1.2 Building a Library with FCS Routines Included

The command files in this section build 8K-word OTS resident libraries that include FCS modules. Note that you must edit F4PRES.MAC in order to bring the size of the library down to 8K words.

The following command file builds an 8K-word OTS resident library (F4PRES.TSK) that includes FCS modules referenced by the OTS:

```
F4PRES/-HD/LI/-PI,F4PRES/-SP/MA,F4PRES=F4PRES
LB:[1,1]F4POTS/LB
/
STACK=0
UNITS=0
PAR=F4PRES:140000:40000
//
```

Note that the FORTRAN-77 OTS object library (LB:[1,1]F4POTS.OBJ) is referenced in this command file. Be sure that this is an object library that was built with FCS modules during installation.

Use the following Task Builder option when linking a task to this library:

```
LIBR=F4PRES:RO
```

7.1.3 Building a Noncluster Library Linked to FCSFSL

The following command file builds an OTS resident library (F4PRES.TSK) for which FCS routines reside in a separate supervisor-mode library. As supplied, F4PRES.MAC builds a 7K-word library to link to FCSFSL. To make the best use of virtual address space, you can edit F4PRES.MAC (according to instructions that it contains) so that it takes up a full 8K words before you build this library. You can use this configuration only on IAS systems that support supervisor-mode libraries.

```
F4PRES/-HD/LI/-PI,F4PRES/-SP/MA,F4PRES=F4PRES
LB:[1,1]F4POTS/LB
/
STACK=0
UNITS=0
SUPLIB=FCSFSL:SV
PAR=F4PRES:140000:40000
//
```

OTS RESIDENT LIBRARIES

Note that the FORTRAN-77 OTS object library (LB:[1,1]F4POTS.OBJ) is referenced in this command file. Be sure that this is an object library that was built with FCS modules during installation.

Use the following Task Builder option when linking a task to this library:

LIBR=F4PRES:RO

CHAPTER 8

RELEASE NOTES

8.1 FORTRAN-77 OTS WITH RMS-11 SUPPORT

The FORTRAN-77 RMS OTS that is installed on IAS is compatible with RMS V2.0 that ships with IAS Version 3.2. You must task-build these applications using RMS V2.0, since the RMS-OTS is not compatible with RMS V1.8.

8.2 F77 TYPING OF RESULT-GENERIC FUNCTIONS

Programmers should be aware of a new rule in the ANSI X3.9-1978 FORTRAN standard that affects the FORTRAN-77 compiler's behavior when typing result-generic functions. This rule, which is documented in the ANSI X3.9-1978 FORTRAN publication, page 8-5, states:

A type-statement that confirms the type of an intrinsic function whose name appears in the Specific Name column of Table 5 is not required, but is permitted. If a generic function name appears in a type statement, such an appearance is not sufficient by itself to remove the generic properties of that function.

That is to say, the following source program:

```
IMPLICIT INTEGER (A-Z)
INTEGER*4 NINT,I
REAL TEST
I=NINT(TEST)
END
```

does not cause the FORTRAN-77 compiler to automatically select the result type of the NINT function to be INTEGER *4.

The FORTRAN-77 rules governing the selection of the result type of integer-valued intrinsic functions are documented in Section 4.2.4 of the PDP-11 FORTRAN-77 User's Guide and are quoted here for convenience:

A number of intrinsic functions provided by FORTRAN-77 (for example, IFIX) produce integer results from real or double-precision arguments. These intrinsic functions are called "result-generic functions." Because the compiler operates in two different modes, INTEGER*2 mode and INTEGER*4 mode, the system provides two internal versions of each of these integer-producing functions: an INTEGER*2 version and an INTEGER*4 version. Selection of the proper version is made by the compiler mode setting, not, as it does for the other intrinsic functions, on the basis of the data type of arguments in the function reference.

RELEASE NOTES

Therefore, the result type of the NINT function is determined solely by the value of the I4 switch specified. When /NOI4 or /-I4 is specified for source compilation, the result type of the NINT function is INTEGER*2, not INTEGER*4.

8.3 COMPILER FAILS TO GENERATE WARNING DIAGNOSTIC

The compiler fails to generate ERROR-85 (name longer than 6 characters) for the following source program:

```
REAL TOLNUM,TOLSLT
READ (1)TOLNUMTOLSLT
END
```

This problem will be corrected in a future release of PDP-11 FORTRAN-77.

8.4 ADJUSTABLE ARRAY USED IN INVALID CONTEXT

When encountering the following illegal use of an adjustable array, the FORTRAN compiler generates an expected compiler error 40 and an unexpected compiler FATAL*10 diagnostic:

```
SUBROUTINE BBBB
COMMON SIZE
INTEGER SIZE, ABC(SIZE)
END
```

As a temporary solution to this problem, you can modify the source program to avoid both the compiler error 40 and the FATAL*10 diagnostic by declaring the integer array ABE in the subroutine BBBB parameter list.

This problem will be corrected in a future release of PDP-11 FORTRAN-77.

8.5 INTEGER ARRAY REFERENCE USED AS INDEX IN BYTE ARRAY SUBSCRIPT

When the FORTRAN-77 compiler attempts to generate code for BYTE array subscripts and the subscript that is itself an INTEGER array reference, the compiler generates a FATAL*10 diagnostic.

The following source program causes the compiler to generate a FATAL*10 diagnostic:

```
SUBROUTINE EXE(L)
BYTE A1(240),A2(2100)
INTEGER*2 T(100)
DO 4300 L=1,N
  A1(T(L))=A2(T(L))
4300 CONTINUE
END
```

As a workaround to this problem, compile the source program with the /NOOP switch.

This problem will be corrected in a future release of PDP-11 FORTRAN-77.

RELEASE NOTES

8.6 COMMON SUBEXPRESSIONS WITHIN CONDITIONAL STATEMENT

When the FORTRAN-77 compiler attempts to generate optimized code for the following source program, the compiler generates a FATAL*10 diagnostic:

```
      BYTE MINUS,ISGN1
      COMMON /DBUFF/ RBUFF(256)
      COMMON /OUTLN/LUN1
      DATA MINUS /'-'/
      IF(RBUFF(103).LT.0) ISGN1 = MINUS
      IF(RBUFF(104).LT.0) ISGN1 = MINUS
      IF(RBUFF(105).LT.0) ISGN1 = MINUS
      IF (RBUFF(101).EQ.3) WRITE (LUN1,*,ERR=950)
      1 ABS(RBUFF(103)),ABS(RBUFF(104))
      IF (RBUFF(101).EQ.4) WRITE (LUN1,*,ERR=950)
      1 ABS(RBUFF(103)),ABS(RBUFF(104)),
      1 ABS(RBUFF(105))
950  CONTINUE
      END
```

As a workaround to this problem, compile the source program with the /NOOP switch.

This problem will be corrected in a future release of PDP-11 FORTRAN-77.

APPENDIX A
COMPILER TASK-BUILD FILES

A.1 PDP-11 FORTRAN-77 COMPILER TASK-BUILD FILE FOR IAS (F7711D.CMD)

```
[11,1]F77/CP/-FP/RW/MU,[11,36]F77/-SP=[11,36]F7711D.ODL/MP
;
; PDP-11 FORTRAN-77 COMPILER TASK BUILD FILE
;
; PDP-11 FORTRAN-77 VERSION 5.0
;       IAS VERSION 3.2 USING MCR
;
; SUMMARY OF SYSTEM PARAMETERS:
;       USES SYSTEM RESIDENT LIBRARY "SYSRES"
;       24K COMPILER TASK
;       512 WORD STACK
;       39 FRAMES IN EXPRESSION ANALYZER STACK
;       20 FRAMES IN DO/BLOCK IF STATEMENT STACK
;       45 COMMON BLOCKS, MAXIMUM
;       12 RESIDENT PAGES FOR WORKFILE SYSTEM
;
;
; OPTION INPUT
;
TASK      =...F77
LIBR      =SYSRES:RO
POOL      =100
; SP STACK OF 512 WORDS
; STACK MUST NEVER BE LESS THAN 384 WORDS
;
STACK     =512
; F77 COMPILER LOGICAL UNIT ASSIGNMENTS
;       1  COMMAND INPUT
;       2  COMMAND OUTPUT
;       3  .OBJ OUTPUT
;       4  .LST OUTPUT
;       5  .FTN INPUT
;
;       6  COMPILER WORKFILE (RANDOM ACCESS)
;           CAN BE REASSIGNED TO SWAPPING DISK IF AVAILABLE
;           DISK MUST BE MOUNTED AS WRITABLE FILES-11 VOLUME,
;           BUT THE WORKFILE DOES NOT REQUIRE A UFD ON THE VOLUME.
;
;       7  COMPILER TEMP FILES (SEQUENTIAL ACCESS)
;       8  DISK MUST BE MOUNTED AS WRITABLE FILES-11 VOLUME,
;           BUT THE TEMP FILES DO NOT REQUIRE A UFD ON THE VOLUME.
;
;       9  COMPILER MESSAGE TEXT FILE
;
UNITS     =9
ASG       =TI:1,TI:2
ASG       =SY0:6,SY0:7,SY0:8
```

COMPILER TASK-BUILD FILES

```

ASG          =LB0:9
; RESIDENT MEMORY FOR WORKFILE VIRTUAL MEMORY SYSTEM
;
; UNDER IAS WORKFILE RESIDENT MEMORY IS DYNAMICALLY ALLOCATED
; BY THE COMPILER BASED ON A PARAMETER SPECIFIED BY "EXTTSK".
; THIS VALUE MAY BE OVERRIDDEN AT INSTALL BY "INSTALL F77/INC=NNNN".
;
; INCREASING THE NUMBER OF RESIDENT WORKFILE PAGES WILL MAKE THE COMPILER
; RUN FASTER BY REDUCING PAGING I/O, BUT IT DOES NOT AFFECT THE SIZE
; OF THE MAXIMUM SOURCE PROGRAM WHICH CAN BE COMPILED.
;
EXTTSK       =3840
; F77 USES CONTROL SECTION "STACK1" FOR:
;     EXPRESSION ANALYZER STACK DURING PASS 1
;     NAMED COMMON BLOCK DEFINITIONS IN LATER PASSES
;
; AS DEFINED BELOW, STACK1 IS 312(10) WORDS, PROVIDING:
;     312/8           = 39 EXPRESSION ANALYZER STACK FRAMES
;     312/6           = 52 CONTROL SECTIONS
;
;     UP TO 7 CONTROL SECTIONS MAY BE USED FOR
;     COMPILER-GENERATED CODE AND DATA, LEAVING 45
COMMON BLOCKS.
;
EXTSCT       =STACK1:1160
; F77 USES CONTROL SECTION "DOSTK1" FOR:
;     DO STATEMENT NESTING STACK DURING PASS 1
;
; AS DEFINED BELOW, DOSTK1 IS 80(10) WORDS, PROVIDING:
;     80/4            = 20 NESTED DO/BLOCK IF STATEMENTS
;
EXTSCT       =DOSTK1:240
; DEFINE PRINTER WIDTH AND NUMBER OF SOURCE LINES PER LISTING PAGE
; F77 DEFAULT VALUES ARE:
;     55 SOURCE LINES PER PAGE (PLUS 3 LINES OF HEADING)
;     132 COLUMN LINE PRINTER
; NOTE:
;     55(10) = 67(8)
;     80(10) = 120(8)  132(10) = 204(8)
;
GBLPAT       =FORTRN:LPLINE:67
GBLPAT       =FORTRN:LPWIDTH:204
; DEFINE DEFAULT OUTPUT FILE SUPERCEDE BEHAVIOR: A VALUE OF 0 (DEFAULT)
; INDICATES THAT THE COMPILER SHOULD NOT SUPERCEDE OUTPUT LISTING AND
; OBJECT FILES; A VALUE OF 1 ALLOWS SUPERCEDING.
;
GBLPAT       =COMAND:SUP00:0
; DEFINE I- AND D-SPACE SUPPORT FOR RSX-11M-PLUS V2.1: A VALUE OF 1
; INDICATES THAT OBJECT MODULES OUTPUT BY THE COMPILER CAN BE USED
; FOR BUILDING I- AND D-SPACE TASKS; A VALUE OF 0 (DEFAULT) DOES NOT
; ALLOW THIS. THE VALUE MUST BE 0 IF FORTRAN-77 DEBUG IS USED.
; THE VALUE MUST BE 0 FOR IAS
;
GBLPAT       =FORTRN:DSPACE:0
; DEFINITION OF COMPILER SWITCH OPTION VALUES
;
; A COMPLETE DESCRIPTION OF THE EFFECTS OF THE COMPILER OPTION SWITCHES
; IS CONTAINED IN SECTION 1.2 OF THE PDP-11 FORTRAN-77 USER'S GUIDE.
;
; SWITCH      SWITCH      VALUE TO GBLPAT
; NAME        SETTING
; -----
; CK          /-CK 0
;            /CK 1 ARRAY SUBSCRIPT BOUNDS CHECKING

```

COMPILER TASK-BUILD FILES

```

;
; CO                                /CO:19. 23 NUMBER OF CONTINUATION LINES
;                                /CO:N.  N
;
; DE                                /-DE 0
;                                /DE 1 INCLUDE DEBUG LINES
;
; I4                                /-I4 0 DEFAULT INTEGER2
;                                /I4 1 DEFAULT INTEGER4
;
; LA                                /-LA 0 REINITIALIZE SWITCHES
;                                /LA 1
;
; LI                                /LI:0 0
;                                /LI:1 1 SOURCE
;                                /LI:2 2 SOURCE, MAP
;                                /LI:3 3 SOURCE, MAP, GENERATED CODE
;
; RO                                /-RO 0 R/W CODE SECTIONS
;                                /RO 1 R/O CODE SECTIONS
;
; SP                                /-SP 0 NO SPOOLING
;                                /SP 1 SPOOLING
;
; TR                                /-TR 0
;                                /TR:NONE 0
;                                /TR:NAMES 1
;                                /TR:BLOCKS 3
;                                /TR:ALL 7
;                                /TR 7
;
; WF                                /WF:2 2 NUMBER OF TEMPORARY FILES
;                                /WF:N 1,2,3
;
; WR                                /-WR 0 NO OPTIONAL WARNINGS
;                                /WR 1
;
; F77                              /-F77 0 FORTRAN 66 INTERPRETATION
;                                /F77 1 FORTRAN 77 INTERPRETATION
;
; ST                                /-ST 0
;                                /ST:NONE 0
;                                /ST:SOURCE 1
;                                /ST:SYNTAX 2
;                                /ST 2
;                                /ST:ALL 3
;
; DB                                /-DB 0 NO DEBUG INFORMATION
;                                /DB 1 PRODUCE DEBUG INFORMATION
;
; EX                                /-EX 0 72 COLUMN PER SOURCE LINE
;                                /EX 1 132 COLUMN PER SOURCE LINE
;
; OP                                /-OP 0 NO CODE OPTIMIZATION
;                                /OP 3 WITH CODE OPTIMIZATION
;
;
; THE FOLOWING "GBLPAT" DEFINITIONS EFFECT DEFAULTS OF:
;
; /-CK/CO:19./-DB/-DE/-EX/-I4/-LA/LI:2/OP/-RO/-SP/TR:BLOCKS/WF:2/WR/F77/-ST
;
; DEFAULT VALUES FOR SWITCH "XX" ARE DEFINED
; BY A "GBLPAT" TO GLOBAL VARIABLE "XX000".

```

COMPILER TASK-BUILD FILES

```

GBLPAT      =F0RTRN:LA000:0
GBLPAT      =COMAND:CK000:0
GBLPAT      =COMAND:CO000:23
GBLPAT      =COMAND:DE000:0
GBLPAT      =COMAND:I4000:0
GBLPAT      =COMAND:LI000:2
GBLPAT      =COMAND:RO000:0
GBLPAT      =COMAND:SP000:1
GBLPAT      =COMAND:TR000:3
GBLPAT      =COMAND:WF000:2
GBLPAT      =COMAND:WR000:1
GBLPAT      =COMAND:F7700:1
GBLPAT      =COMAND:ST000:0
GBLPAT      =COMAND:DB000:0
GBLPAT      =COMAND:EX000:0
GBLPAT      =COMAND:OP000:3
/
/TASK:[11,1]F77/MAP:[11,36]F77/OVERLAY:[11,36]F7711D-
/OPTIONS/MULTIUSER/NOFLOATING/CHECKPOINT/READWRITE
!
! PDP-11 FORTRAN-77 COMPILER TASK BUILD FILE
!
! PDP-11 FORTRAN-77 VERSION 5.0
!       IAS VERSION 3.2 USING PDS
!
! SUMMARY OF SYSTEM PARAMETERS:
!       USES SYSTEM RESIDENT LIBRARY "SYSRES"
!       24K COMPILER TASK
!       512 WORD STACK
!       39 FRAMES IN EXPRESSION ANALYZER STACK
!       20 FRAMES IN DO/BLOCK IF STATEMENT STACK
!       45 COMMON BLOCKS, MAXIMUM
!       12 RESIDENT PAGES FOR WORKFILE SYSTEM
!
! OPTION INPUT
!
TASK         =...F77
LIBR         =SYSRES:RO
POOL         =100
!
! SP STACK OF 512 WORDS
! STACK MUST NEVER BE LESS THAN 384 WORDS
!
STACK        =512
!
! F77 COMPILER LOGICAL UNIT ASSIGNMENTS
!       1  COMMAND INPUT
!       2  COMMAND OUTPUT
!       3  .OBJ OUTPUT
!       4  .LST OUTPUT
!       5  .FTN INPUT
!
!       6  COMPILER WORKFILE (RANDOM ACCESS)
!           CAN BE REASSIGNED TO SWAPPING DISK IF AVAILABLE
!           DISK MUST BE MOUNTED AS WRITABLE FILES-11 VOLUME,
!           BUT THE WORKFILE DOES NOT REQUIRE A UFD ON THE VOLUME.
!
!       7  COMPILER TEMP FILES (SEQUENTIAL ACCESS)
!       8  DISK MUST BE MOUNTED AS WRITABLE FILES-11 VOLUME,
!           BUT THE TEMP FILES DO NOT REQUIRE A UFD ON THE VOLUME.
!
!       9  COMPILER MESSAGE TEXT FILE
!

```


COMPILER TASK-BUILD FILES

```

UNITS          =9
ASG            =TI:1,TI:2
ASG            =SP0:6,SP0:7,SP0:8
ASG            =LB0:9
!
! RESIDENT MEMORY FOR WORKFILE VIRTUAL MEMORY SYSTEM
!
! UNDER IAS WORKFILE RESIDENT MEMORY IS DYNAMICALLY ALLOCATED
! BY THE COMPILER BASED ON A PARAMETER SPECIFIED BY "EXTTSK".
! THIS VALUE MAY BE OVERRIDDEN AT INSTALL BY "INSTALL/INC:NNNN [11,1]F77".
!
! INCREASING THE NUMBER OF RESIDENT WORKFILE PAGES WILL MAKE THE COMPILER
! RUN FASTER BY REDUCING PAGING I/O, BUT IT DOES NOT AFFECT THE SIZE
! OF THE MAXIMUM SOURCE PROGRAM WHICH CAN BE COMPILED.
!
EXTTSK         =3840
!
! F77 USES CONTROL SECTION "STACK1" FOR:
!           EXPRESSION ANALYZER STACK DURING PASS 1
!           NAMED COMMON BLOCK DEFINITIONS IN LATER PASSES
!
! AS DEFINED BELOW, STACK1 IS 312(10) WORDS, PROVIDING:
!           312/8           = 39 EXPRESSION ANALYZER STACK FRAMES
!           312/6           = 52 CONTROL SECTIONS
!                           UP TO 7 CONTROL SECTIONS MAY BE USED FOR
!                           COMPILER-GENERATED CODE AND DATA, LEAVING 45
COMMON BLOCKS.
!
EXTSCT         =STACK1:1160
!
! F77 USES CONTROL SECTION "DOSTK1" FOR:
!           DO STATEMENT NESTING STACK DURING PASS 1
!
! AS DEFINED BELOW, DOSTK1 IS 80(10) WORDS, PROVIDING:
!           80/4            = 20 NESTED DO/BLOCK IF STATEMENTS
!
EXTSCT         =DOSTK1:240
!
! DEFINE PRINTER WIDTH AND NUMBER OF SOURCE LINES PER LISTING PAGE
! F77 DEFAULT VALUES ARE:
!           55 SOURCE LINES PER PAGE (PLUS 3 LINES OF HEADING)
!           132 COLUMN LINE PRINTER
! NOTE:
!           55(10) = 67(8)
!           80(10) = 120(8)  132(10) = 204(8)
!
GBLPAT         =FORTRN:LPLINE:67
GBLPAT         =FORTRN:LPWIDTH:204
!
! DEFINE DEFAULT OUTPUT FILE SUPERCEDE BEHAVIOR: A VALUE OF 0 (DEFAULT)
! INDICATES THAT THE COMPILER SHOULD NOT SUPERCEDE OUTPUT LISTING AND
! OBJECT FILES! A VALUE OF 1 ALLOWS SUPERCEDING.
!
GBLPAT         =COMAND:SUP00:0

; DEFINE I- AND D-SPACE SUPPORT FOR RSX-11M-PLUS V2.1: A VALUE OF 1
; INDICATES THAT OBJECT MODULES OUTPUT BY THE COMPILER CAN BE USED
; FOR BUILDING I- AND D-SPACE TASKS; A VALUE OF 0 (DEFAULT) DOES NOT
; ALLOW THIS. THE VALUE MUST BE 0 IF FORTRAN-77 DEBUG IS USED.
; THE VALUE MUST BE 0 FOR IAS
;
GBLPAT         =FORTRN:DSPACE:0

; DEFINITION OF COMPILER SWITCH OPTION VALUES

```

COMPILER TASK-BUILD FILES

```

;
; A COMPLETE DESCRIPTION OF THE EFFECTS OF THE COMPILER OPTION SWITCHES
; IS CONTAINED IN SECTION 1.2 OF THE PDP-11 FORTRAN-77 USER'S GUIDE.
;
; SWITCH      SWITCH      VALUE TO GBLPAT
; NAME        NAME        SETTING
; -----
;
; CK          /-CK 0
;             /CK 1 ARRAY SUBSCRIPT BOUNDS CHECKING
;
; CO          /CO:19. 23 NUMBER OF CONTINUATION LINES
;             /CO:N.  N
;
; DE          /-DE 0
;             /DE 1 INCLUDE DEBUG LINES
;
; I4          /-I4 0 DEFAULT INTEGER2
;             /I4 1 DEFAULT INTEGER4
;
; LA          /-LA 0 REINITIALIZE SWITCHES
;             /LA 1
;
; LI          /LI:0 0
;             /LI:1 1 SOURCE
;             /LI:2 2 SOURCE, MAP
;             /LI:3 3 SOURCE, MAP, GENERATED CODE
;
; RO          /-RO 0 R/W CODE SECTIONS
;             /RO 1 R/O CODE SECTIONS
;
; SP          /-SP 0 NO SPOOLING
;             /SP 1 SPOOLING
;
; TR          /-TR 0
;             /TR:NONE 0
;             /TR:NAMES 1
;             /TR:BLOCKS 3
;             /TR:ALL 7
;             /TR 7
;
; WF          /WF:2 2 NUMBER OF TEMPORARY FILES
;             /WF:N 1,2,3
;
; WR          /-WR 0 NO OPTIONAL WARNINGS
;             /WR 1
;
; F77         /-F77 0 FORTRAN 66 INTERPRETATION
;             /F77 1 FORTRAN 77 INTERPRETATION
;
; ST          /-ST 0
;             /ST:NONE 0
;             /ST:SOURCE 1
;             /ST:SYNTAX 2
;             /ST 2
;             /ST:ALL 3
;
; DB          /-DB 0 NO DEBUG INFORMATION
;             /DB 1 PRODUCE DEBUG INFORMATION
;
; EX          /-EX 0 72 COLUMN PER SOURCE LINE
;             /EX 1 132 COLUMN PER SOURCE LINE
;

```

COMPILER TASK-BUILD FILES

```

; OP                      /-OP  0 NO CODE OPTIMIZATION
; OP                      /OP  3 WITH CODE OPTIMIZATION
;
;
; THE FOLOWING "GBLPAT" DEFINITIONS EFFECT DEFAULTS OF:
;
; /-CK/CO:19./-DB/-DE/-EX/-I4/-LA/LI:2/OP/-RO/-SP/TR:BLOCKS/WF:2/WR/F77/-ST
;
; DEFAULT VALUES FOR SWITCH "XX" ARE DEFINED
; BY A "GBLPAT" TO GLOBAL VARIABLE "XX000".

```

```

GBLPAT      =FOTRN:LA000:0
GBLPAT      =COMAND:CK000:0
GBLPAT      =COMAND:CO000:23
GBLPAT      =COMAND:DE000:0
GBLPAT      =COMAND:I4000:0
GBLPAT      =COMAND:LI000:2
GBLPAT      =COMAND:RO000:0
GBLPAT      =COMAND:SP000:1
GBLPAT      =COMAND:TR000:3
GBLPAT      =COMAND:WF000:2
GBLPAT      =COMAND:WR000:1
GBLPAT      =COMAND:F7700:1
GBLPAT      =COMAND:ST000:0
GBLPAT      =COMAND:DB000:0
GBLPAT      =COMAND:EX000:0
GBLPAT      =COMAND:OP000:3
/

```

