

PDP-11 FORTRAN-77/RT-11

Installation Guide/Release Notes

AA-BR72A-TC

March 1984

This guide describes the procedures for installing PDP-11 FORTRAN-77 on the RT-11 operating system.

SUPERSESSION/UPDATE INFORMATION:	This is a new manual for this release.
OPERATING SYSTEM AND VERSION:	RT-11 V5.1
SOFTWARE VERSION:	FORTRAN-77/RT-11 V5.0



Adapted from PDP-11 FORTRAN-77/RSX by Multiware, Inc.
139 G Street, Suite 161, Davis, California 95616

digital equipment corporation • maynard, massachusetts

First Printing, March 1984

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 be used or copied only 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 Equipment Corporation or its affiliated companies.

Copyright © 1984 by Digital Equipment Corporation
All Rights Reserved

Printed in U.S.A.

The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECTape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EduSystem	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FOCAL	RSX
COMTEX	INDAX	TYPESET-8
DDT	LAB-8	TYPESET-11
DECCOMM	DECSYSTEM-20	TMS-11
ASSIST-11	RTS-8	ITPS-10
VAX	VMS	SBI
DECnet	IAS	PDT
DATATRIEVE	TRAX	digital

MW-F77-006

HOW TO ORDER ADDITIONAL DOCUMENTATION

In Continental USA and Puerto Rico call 800-258-1710

In New Hampshire, Alaska, and Hawaii call 603-884-6660

In Canada call 613-234-7726 (Ottawa-Hull)
800-267-6146 (all other Canadian)

DIRECT MAIL ORDERS (USA & PUERTO RICO)*

Digital Equipment Corporation
P.O. Box CS2008
Nashua, New Hampshire 03061

DIRECT MAIL ORDERS (CANADA)

Digital Equipment of Canada Ltd.
940 Belfast Road
Ottawa, Ontario K1G 4C2
Attn: A&SG Business Manager

DIRECT MAIL ORDERS (INTERNATIONAL)

Digital Equipment Corporation
A&SG Business Manager
c/o Digital's local subsidiary or
approved distributor

*Any prepaid order from Puerto Rico must be placed with the local Digital subsidiary (809-754-7575)

International orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Northboro, Massachusetts 01532.

Table of Contents

CHAPTER 1	INTRODUCTION	
1.1	SYSTEM REQUIREMENTS	1-1
1.2	DISTRIBUTION FILES	1-2
CHAPTER 2	INSTALLATION	
2.1	BACK UP THE DISTRIBUTION KIT	2-1
2.2	DECIDE WHAT PART OF THE DISTRIBUTION YOU NEED	2-1
2.3	FREE UP AVAILABLE MEMORY	2-2
2.3.1	Removing SPOOL and VTCOM	2-2
2.3.2	Removing the Single-Line Editor	2-3
2.3.3	Removing unnecessary handlers	2-4
2.3.4	Swapping the USR	2-4
2.3.5	Making free memory contiguous	2-4
2.4	LOAD AND ASSIGN THE DISTRIBUTION VOLUME	2-4
2.5	MAKE ROOM ON THE SYSTEM DEVICE	2-5
2.6	COPY REQUIRED FORTRAN-77 FILES FROM THE DISTRIBUTION KIT	2-5
2.7	PATCH THE LINKER	2-6
2.8	RUN THE INSTALLATION VERIFICATION PROGRAM	2-7
2.8.1	Compile the test program	2-7
2.8.2	Link the test program	2-8
2.8.3	Run the test program	2-8
2.9	COPY OPTIONAL OTS MODULES TO THE SYSTEM	2-9
2.10	INSTALL THE OTS IN SYSLIB	2-9
2.10.1	SYSUPD	2-10
2.10.2	Installing OTS in SYSLIB	2-10
2.11	RENAME THE COMPILER	2-10
2.12	ERRORS	2-11
CHAPTER 3	OPTIONAL MODULES	
3.1	VIRTXM	3-1
3.2	SHORT ERROR TEXT	3-1
3.3	F77MAP	3-1
3.4	F77EIS	3-2
3.5	F77CVF	3-2
3.6	F77NER	3-2
3.7	F77RAN	3-2
CHAPTER 4	OPTIONAL INSTALLATION PROCEDURE	
4.1	STEPS OF THE OPTIONAL PROCEDURE	4-1
4.2	RUNNING THE IND COMMAND FILE	4-1
4.2.1	Steps Common to all Distribution Media	4-3
4.3	RUNNING THE VERIFICATION TEST	4-5
4.4	MOVING VIRTXM TO THE SYSTEM	4-5

CHAPTER 5	SYSTEM PERFORMANCE AND TAILORING	
5.1	FACTORS AFFECTING COMPILER PERFORMANCE	5-1
5.2	COMPILER TAILORING	5-2
5.2.1	PSECT SWITCH	5-3
5.2.2	PSECT WRKNAM	5-3
5.2.3	Modifying the compiler image	5-3
CHAPTER 6	RELEASE NOTES	
6.1	VIRTUAL ARRAYS NOT SUPPORTED UNDER FB AND SJ MONITORS	6-1
6.2	USING AN UNPATCHED LINKER	6-1
6.3	UNFORMATTED SEQUENTIAL ARRAYS	6-1
6.4	OTSHK1	6-1
6.5	NON-FILE STRUCTURED LOOKUPS	6-2
6.6	MIXING COMMON DATA TYPES	6-2
6.7	F77 TYPING OF RESULT-GENERIC FUNCTIONS	6-2
6.8	COMPILER FAILS TO GENERATE WARNING DIAGNOSTIC	6-3
6.9	ADJUSTABLE ARRAY USED IN INVALID CONTEXT	6-3
6.10	INTEGER ARRAY REFERENCE USED AS INDEX IN BYTE ARRAY SUBSCRIPT	6-3
6.11	COMMON SUBEXPRESSIONS WITHIN CONDITIONAL STATEMENT	6-3
APPENDIX A	COMPILER PSECTS THAT CAN BE MODIFIED	

PREFACE

MANUAL OBJECTIVE

Tell how to install the PDP-11 FORTRAN-77/RT-11 compiler and Object Time System (OTS) in the RT-11 operating system.

INTENDED AUDIENCE

This manual is intended for use by RT-11 system managers and FORTRAN programmers.

STRUCTURE OF THIS DOCUMENT

- Chapter 1 describes the system requirements for the installation and use of PDP-11 FORTRAN-77/RT-11, and lists the files to be found in the distribution kit.
- Chapter 2 provides step-by-step instructions for installing the compiler and OTS from the distribution kit.
- Chapter 3 discusses various optional OTS modules in the distribution kit.
- Chapter 4 describes use of the optional IND command file for installation assistance.
- Chapter 5 discusses factors affecting compiler performance and ways of customizing the compiler.
- Chapter 6 contains release notes.
- Appendix A gives compiler PSECT information that is useful for customizing the compiler.

ASSOCIATED DOCUMENTS

You should be familiar with the RT-11 operating system before attempting installation. The PDP-11 FORTRAN-77/RT-11 User's Guide provides detailed information on using the FORTRAN-77/RT-11 system. The PDP-11 FORTRAN-77 Language Reference Manual describes the elements of the FORTRAN language as implemented in PDP-11 FORTRAN-77/RT-11. For a complete list of operating system software documents, see the RT-11 operating-system documentation directory.

CONVENTIONS USED IN THIS MANUAL

The following conventions are used in this manual:

CTRL/x or ^x	The notation CTRL/x or ^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 indicate information that must be entered exactly 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 describes the features you must have on your PDP-11 or Professional 300 series system to install and use PDP-11 FORTRAN-77/RT-11, and how you can verify that your system has these features. Section 1.2 lists the files contained on the PDP-11 FORTRAN-77/RT-11 distribution kit.

1.1 SYSTEM REQUIREMENTS

The following hardware is needed to install and use PDP-11 FORTRAN-77/RT-11:

- A PDP-11 or Professional 300 series processor equipped with an FP-11 type floating-point processor.
- RT-11, Version 5.0 or 5.1. If your processor is one of the Professional 300 series, then you must have the RT11XM monitor installed.
- At least 56KB of memory for the SJ and FB versions of the RT-11 monitor, or at least 256KB for the XM version of the monitor.
- At least 670 free blocks of disk space.
- A KT-11 memory management unit if you intend to use virtual arrays.

You can get a description of your system hardware and software by typing

```
SHOW CONFIGURATION<RET>
```

Examine the information displayed to confirm that you have the correct hardware and software.

NOTE: PDP-11 FORTRAN-77/RT-11 can be used on systems without an FP11 floating-point unit if the application requires no floating-point calculations and the optional module F77EIS.OBJ, described in Chapter 3, is linked with the job.

INTRODUCTION

1.2 DISTRIBUTION FILES

The PDP-11 FORTRAN-77/RT-11 distribution kit for the RT-11 operating system is available on floppy disk, magnetic tape and disk cartridge media. These media contain the following files:

File Name Description

INSTAL.COM	Command procedure that performs the installation.
F77.SAV	FORTTRAN-77 Compiler for SJ and FB monitors.
F77XM.SAV	FORTTRAN-77 Compiler for XM monitor.
F77COM.MSG	Compiler diagnostic messages file.
F77OTS.OBJ	PDP-11 FORTRAN-77/RT-11 compiler object module library.
SHORT.OBJ	OTS short-error text module.
F77MAP.OBJ	PDF name-mapping concatenated object module.
F77CVF.OBJ	Optional OTS floating-point formatted output conversion routine.
F77EIS.OBJ	EIS replacements for OTS integer functions that make use of the FPP.
F77NER.OBJ	Optional OTS error-reporting module.
F77RAN.OBJ	Optional random-number generator.
F77TST.FOR	System installation verification test program.
VIRTXM.OBJ	Module that must be linked with programs that use virtual arrays under the XM monitor.
SYSUPD.OBJ	Concatenated object module containing replacements for FORTRAN-IV specific SYSLIB routines to make them compatible with FORTRAN-77.

CHAPTER 2

INSTALLATION

This chapter tells you how to install PDP-11 FORTRAN-77/RT-11 from the distribution kit. Distribution kits are available for RX50 floppy disks, RX01 floppy disks, RL01 removable rigid disks, and RT-11 structured magnetic tape.

The basic installation procedure consists of the following steps:

- Back up the distribution kit.
- Decide what part of the distribution kit you need.
- Free up available memory.
- Load and assign the distribution volume.
- Make room on the system device.
- Copy required files from the distribution kit.
- Patch the linker.
- Run the Installation Verification program.
- Copy optional OTS modules from the distribution kit.
- Install the OTS in SYSLIB.
- Rename the compiler.

An optional IND command file, INSTAL.COM, is available on the distribution kit to assist you in the installation process. If you would like to use this optional procedure, see Chapter 4.

2.1 BACK UP THE DISTRIBUTION KIT

If it is possible to make a copy of the distribution kit, you should do so before proceeding with the installation. This enables you to file the original distribution kit safely away and perform the actual installation with the copy.

2.2 DECIDE WHAT PART OF THE DISTRIBUTION YOU NEED

The distribution kit contains two versions of the compiler. F77.SAV is for use with the SJ and FB monitors, and F77XM.SAV is for use with the XM monitor. Determine which version of the compiler you need based on the type of RT monitor you have. If you have both the XM monitor and one or both of the other monitors on your system, you may

INSTALLATION

want to install both versions of the compiler. If you have a Professional 300 series system, however, you should use only the XM version of the compiler, which must be run under the XM monitor.

The distribution kit also contains some optional object modules you might want to put on your system, although none of them is required. These modules are described in Chapter 3.

2.3 FREE UP AVAILABLE MEMORY

Some parts of the installation procedure require a fairly large amount of free memory. If you do not have enough, you will not be able to complete the installation. For example, running the XM version of the compiler requires at least 12.5K words of background memory and 58K words of extended memory be available. To ensure there is enough free memory available for the installation, you should temporarily abort foreground and system jobs, and unload handlers that are not needed for the installation procedure. The jobs may be restarted and the handlers loaded again after the installation is complete.

Use the RT-11 SHOW MEMORY and SHOW JOBS commands to see what is loaded and running on your system. Make a note of the amount of memory that is listed as unused. Background memory is indicated by "..BG..", and free extended memory by "....." under the Extended Memory heading (XM monitor only). Compare the amount of contiguous background and extended memory with that listed as required in Table 2-1. If there is adequate memory available, you can skip ahead now to section 2.4. Otherwise, you must take steps towards maximizing available system memory. The following sections can help you with this task.

Table 2-1
Required Memory for FORTRAN-77 Installation

Memory Area	SJ or FB Monitor	XM Monitor
Background (BG) Memory	21500. words	12500. words
Extended Memory	<none>	58000. words

Note that an occurrence of "....." under the "Kernal Memory" heading denotes a discontinuity in the background area. See section 2.3.5 for information on how to gather these areas into the background.

2.3.1 Removing SPOOL and VTCOM

Temporarily removing the spooler and the remote communications program can free up a lot of memory. These programs are not needed for installing or running the FORTRAN-77 compiler. If either of these programs are currently loaded and running, you should remove them now.

The SPOOL utility can be removed by typing the following sequence:

```
ABORT SPOOL<RET>  
UNLOAD SPOOL<RET>  
UNLOAD SP<RET>
```

The VTCOM utility can be removed by typing the command sequence on the left. The comments on the right explain each step:

INSTALLATION

```
^X                ! control-X awakes system-job prompt
JOB> VTCOM<RET>   ! direct input to VTCOM
F> ^P            ! control-P requests VTCOM prompt
TT::VTCOM> EX<RET> ! type the EXIT command
B>              ! RT-11 indicates that Background is
                ! now accepting input
UNLOAD VTCOM<RET> ! Free the space occupied by VTCOM.
UNLOAD XC<RET>   ! if computer is PRO-300 series, or
UNLOAD XL<RET>   ! if other PDP-11.
```

Later, if these programs are needed, you can load and run them as required. If the computer is a Professional 300 series, or a PDP-11 that uses a serial line printer interface, you can restart the spooler by typing

```
SRUN SY:SPOOL/PAUSE<RET>
LOAD LS=SPOOL<RET>
RESUME SPOOL<RET>
```

Otherwise, if you have a PDP-11 with a parallel printer interface, type

```
SRUN SY:SPOOL/PAUSE<RET>
LOAD LP=SPOOL<RET>
RESUME SPOOL<RET>
```

The VTCOM program can be restarted by typing the following sequence if the computer is a Professional 300 series:

```
FRUN SY:VTCOM.SAV/PAUSE<RET>
LOAD XC=VTCOM<RET>
RESUME VTCOM<RET>
```

If the computer is a PDP-11, type

```
FRUN SY:VTCOM.SAV/PAUSE<RET>
LOAD XL=VTCOM<RET>
RESUME VTCOM<RET>
```

In general, when you use large background programs such as the FORTRAN-77 compiler, you will have to unload these auxiliary system jobs to maximize the available system memory. Again, use the SHOW MEMORY command to see if enough memory has been freed. If so, skip to section 2.4. If not, you will have to remove other modules.

2.3.2 Removing the Single-Line Editor

The Single-Line Editor is another optional system program that can be removed if more system memory must be reclaimed. It can be removed by typing

```
SET SL OFF<RET>
UNLOAD SL<RET>
REMOVE SL<RET>
```

After the installation is complete, the Single-Line Editor can be started again by typing

```
INSTALL SL<RET>
LOAD SL<RET>
SET SL ON<RET>
```

INSTALLATION

2.3.3 Removing unnecessary handlers

To find out what handlers can be unloaded, type

```
SHOW<RET>
```

Any handlers you do not need and that are indicated as "(Loaded)" should be unloaded. For example, if XL is loaded, you should type

```
UNLOAD XL:<RET>
```

2.3.4 Swapping the USR

If you are running the Single Job (SJ) or Foreground/Background (FB) monitor, you can easily make available an extra 2K words of memory by allowing the User Service Routine to swap out of memory when not needed. To do this, type

```
SET USR SWAP<RET>
```

Note that the Extended Memory (XM) Monitor does not allow you to SWAP the USR.

2.3.5 Making free memory contiguous

Type

```
SHOW MEMORY<RET>
```

to see how memory is allocated. If the free background or kernel memory is broken up into two or more pieces, you will have to UNLOAD any remaining unloadable handlers or jobs to bring the memory pieces together into one region, then re-LOAD the handlers or jobs that you unloaded. The USR should "slide up" automatically as modules that lie above it are removed.

2.4 LOAD AND ASSIGN THE DISTRIBUTION VOLUME

RX50: The RX50 distribution kit consists of two diskettes. Volume one contains all the files necessary for installation on SJ and FB monitors, while volume two has all the files needed for installation on an XM monitor. (If you have a Professional 300 series system, you must use the XM monitor, so you will need only volume two.) Select the diskette that corresponds to the type of monitor you are using, put a WRITE PROTECT tab on it if it doesn't have one, then load it in an available floppy drive.

RL01: Install the distribution disk in an available RL01 drive. Be sure to WRITE PROTECT or WRITE LOCK the disk.

RX01: The RX01 distribution kit contains two diskettes, both of which are required for the installation process. Volume one contains the XM version of the compiler (F77XM.SAV), the FORTRAN-77 OTS library (F77OTS.OBJ), and the optional IND command file (INSTAL.COM). Volume two contains the SJ/FB version of the compiler (F77.SAV), the compiler error message file (F77COM.MSG), and all the optional object modules discussed in Chapter 3. If you have available mass storage elsewhere on your system that can hold the entire contents of the distribution kit, you may find it more convenient to transfer all the files there. Otherwise you will have to be sure the correct diskette is loaded for

INSTALLATION

each step of the installation. If you are going to perform the installation directly from diskette, install volume one in an available RX01 drive.

MAGTAPE: Make sure there is no write ring inserted in the back of the distribution tape, load it on the tape drive, and position it at the LOAD POINT. Since magnetic tape is not a random-access medium, performing the installation directly from the tape will be slow, and the operations involving LIBR will not work correctly. If you have available mass storage elsewhere on your system that can hold the entire contents of the distribution kit, you should transfer the all the files there and use that device for input.

ALL DISTRIBUTIONS: Assign the logical name INP to the physical device where the distribution kit is loaded. For example, if you have a Professional 300 series system, and you loaded the distribution diskette into drive 1 of the RX50 floppy drive (physical device DZ), then you should type

```
ASSIGN DZ1 INP<RET>
```

2.5 MAKE ROOM ON THE SYSTEM DEVICE

Obtain a directory of the distribution kit by typing

```
DIRECTORY INP:<RET>
```

If you are performing the installation directly from RX01 diskettes, you will have to load the second diskette and repeat this command to get a directory of that diskette also.

The directory will show the size of each of the files on the distribution kit. You should total the sizes of all the files you want in order to determine how much disk storage you require. As discussed in section 2.2, you will need F77.SAV or F77XM.SAV (or both). You will also need the files F77OTS.OBJ and F77COM.MSG. These files must all reside on the system disk, so enough free space must be available for them. To find out how much room is available on the system disk, type

```
DIRECTORY/FREE SY:<RET>
```

The directory listing of the system device displayed on your terminal will reflect the available free space on your system disk. You must be sure that there is enough contiguous (all-in-one-chunk) space for each file. For example, F77XM.SAV would require at least 341 free contiguous blocks. If there is adequate space on the system disk but it is broken up into pieces that are not big enough, you will have to use the RT-11 SQUEEZE command to compress the free space on the disk. See the RT-11 System User's Guide for information on the use of the SQUEEZE command. If the system disk does not contain sufficient free space, then you will have to delete some of the files there or move them to some other device to make enough room.

2.6 COPY REQUIRED FORTRAN-77 FILES FROM THE DISTRIBUTION KIT

NOTE: If you are performing the installation directly from RX01 media, you may have to change diskettes during this step.

INSTALLATION

Copy the desired compiler(s) from the distribution media to the system disk by typing

```
COPY INP:F77.SAV SY:<RET>           for the SJ/FB version,  
or  
COPY INP:F77XM.SAV SY:<RET>        for the XM version.
```

Copy the FORTRAN-77 Object Time System Library (OTS) and the compiler diagnostic messages file to the system device with the commands

```
COPY INP:F77OTS.OBJ SY:<RET>  
COPY INP:F77COM.MSG SY:<RET>
```

2.7 PATCH THE LINKER

There is an error in the distributed version of the RT-11 linker for versions before V08.04 that occasionally causes the linker to terminate abnormally, displaying an "Invalid GSD" error message when processing certain object files produced by the FORTRAN-77 compiler. The problem has been eliminated for linker versions V08.04 and later. If your linker is version V08.01 or V08.03, the patch described below will repair this defect.

If your linker version is earlier than V08.01, the patch will not fix the LINK utility correctly and should not be applied. You can use the FORTRAN-77 compiler in this case, but you may not be able to link certain object modules. See the release notes in Chapter 6 for methods of minimizing this problem.

To determine the version of your linker, first type

```
R LINK<RET>
```

The linker will respond with an asterisk (*) prompt. Type another <RET>, and the linker will respond with its version number and another prompt. Type CTRL/C to leave the linker.

If your linker version is V08.01 or V08.03, then you should use the Save Image Patch Program (SIPP) to patch the linker. (For complete information on the SIPP commands, see Chapter 20 of the RT-11 System Utilities Manual.)

A description of how to apply the patch to the linker follows.

Type

```
COPY/PROTECT SY:LINK.SAV SY:LINK.DIS<RET>  
UNPROTECT SY:LINK.SAV<RET>  
R SIPP<RET>
```

If the system reports that it cannot find SIPP.SAV, then you must locate it on your RT-11 distribution media, copy it to your system device, and retype R SIPP<RET>.

If you get the SIPP prompt (*), then type the responses shown (bold text is SIPP output):

```
*SY:LINK.SAV/A<RET>  
Base? 20462<RET>  
Offset? 0<RET>  
Base Offset Old New?  
020462 000000 173572 173756<RET>  
020462 000002 012604 ^Y<RET>  
*^C
```

INSTALLATION

If an error was reported, or the numbers under the "Old" column above were different than those printed on your terminal, then the patch was not correctly applied and the original version of the linker should be restored by renaming LINK.DIS back to LINK.SAV

If no error was reported, then you have correctly modified the linker. If you are absolutely certain that you will not need the original unpatched version of the linker, you may delete it by typing

```
DELETE SY:LINK.DIS/NOQ<RET>
```

In either case, the linker should be protected by typing

```
PROTECT SY:LINK.SAV<RET>
```

2.8 RUN THE INSTALLATION VERIFICATION PROGRAM

The distribution kit includes a simple test program to verify that PDP-11 FORTRAN-77/RT-11 is correctly installed and ready for use.

NOTE

The test program, F77TST.FOR, contains two intentional non-fatal errors to test the error-reporting facilities of FORTRAN-77. One of the errors occurs during compilation and the other during execution.

Copy the verification test program from the distribution media to your default device by typing

```
COPY INP:F77TST.FOR DK:<RET>
```

2.8.1 Compile the test program

If you are running the SJ or FB version of the RT-11 monitor, compile the test program by typing

```
R F77
*F77TST=F77TST/F:30./S:ALL<RET>
*^C
```

If you are running the XM version of the RT-11 monitor, compile the test program by typing

```
R F77XM
*F77TST=F77TST/F:30./S:ALL<RET>
*^C
```

Wait a few seconds for the compilation to complete. If the compilation is successful, the following error will be reported:

```
F77 -- ERROR 28-E Overflow while converting constant ...
      [ I = 71347] in module F77TST at line 6
F77 -- 1 error
```

This error occurs because the verification program purposely has an invalid statement in it so that correct error reporting can be verified.

INSTALLATION

If this error message is incomplete, then the compiler error-message file, F77COM.MSG, could not be found by the compiler on the system device. In this case, the compiler treats the compile-time overflow in F77TST as a fatal error, and does not create the object module F77TST.OBJ.

If you got the message

```
?KMON-F-Insufficient memory
```

or

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

then there was not enough free background memory available for the compiler. Unload unneeded handlers and stop system and background jobs to temporarily make more background memory available, and try the compilation again.

If you got the message

```
F77 -- FATAL 01 * Open error on work file
```

then there was not enough space on the default device for the compiler to open the temporary (scratch) work file that it uses to build internal tables. Free up at least 128 contiguous blocks of space on the default device by eliminating unwanted files or performing a SQUEEZE on the device. Then try compiling F77TST again.

2.8.2 Link the test program

To link the test program with the FORTRAN-77 OTS, type

```
LINK F77TST,SY:F77OTS<RET>
```

Linking will take a few seconds to complete.

No errors should occur during linking unless the compiler did not create an object module or could not find the file F77OTS.OBJ on the system device.

2.8.3 Run the test program

To execute the test program, type

```
RUN F77TST<RET>
```

If the test program executes successfully, the following message will appear:

```
ERROR 73
Floating zero divide
at PC = 001070
  in "F77TST" at 8
PDP-11 FORTRAN-77 INSTALLATION SUCCESSFUL!
STOP
```

If your system does not have a floating-point processor, the following run-time error is printed:

```
Exiting due to ERROR 4
Segment fault
at PC = 001004
```


INSTALLATION

You have now completed all mandatory steps to the installation procedure. It is highly recommended that you read further in this chapter about the optional installation steps. They will allow you to

- use VIRTUAL arrays in your programs
- use the short error text module to maximize run-time memory
- use SYSLIB modules IASIGN, ILUN, SECNDS, PUTSTR and GETSTR
- combine the FORTRAN-77 OTS with RT-11 System Library (SYSLIB) for simpler LINK commands
- use the Keyboard Monitor's FORTRAN command

If none of these capabilities is desired, you can begin using the FORTRAN-77 system on RT-11. You are encouraged, however, to read on about the optional installation steps.

2.9 COPY OPTIONAL OTS MODULES TO THE SYSTEM

If you want to put some of the optional OTS modules described in Chapter 3 on your system, you should copy them from the distribution at this time. You do not need to copy any modules you do not intend to use, but you may want to copy the module VIRTXM.OBJ, required for linking with any FORTRAN-77 program that uses VIRTUAL arrays. The following command shows how the optional module SHORT.OBJ can be transferred to the device RK3:

```
COPY INP:SHORT.OBJ RK3:<RET>
```

2.10 INSTALL THE OTS IN SYSLIB

FORTTRAN-77 programs must be linked with the FORTRAN-77 OTS library F77OTS.OBJ. If you intend to use FORTRAN-77 a lot, you may find it convenient to add the FORTRAN-77 OTS to your RT-11 system library SYSLIB.OBJ so that you won't have to name the FORTRAN-77 OTS library every time you link a FORTRAN-77 program. You won't be able to do this, however, if you also intend to use RT-11 FORTRAN-IV on your system and you want the FORTRAN-IV OTS to be part of SYSLIB, because the FORTRAN-77 OTS and the FORTRAN-IV OTS cannot coexist in the same library. Decide whether you want the FORTRAN-77 OTS, the FORTRAN-IV OTS, or neither one to be included in SYSLIB.

If you plan to modify SYSLIB, you should make a backup copy of it in case something goes wrong. You can do this by typing

```
COPY/PROTECT SY:SYSLIB.OBJ SY:SYSLIB.OLD<RET>
```

Later, after you have successfully modified SYSLIB, you can delete the old version SYSLIB.OLD.

If your current SYSLIB includes the FORTRAN-IV OTS and you need to remove it, you should load a fresh copy of the RT-11 distributed SYSLIB from your RT-11 installation kit onto your system device. If you have added any modules to the old SYSLIB other than FORTRAN-IV OTS, you should add the same modules to the SYSLIB.

If you have removed the FORTRAN-IV OTS from SYSLIB, but you still intend to use FORTRAN-IV, you will need to put the FORTRAN-IV OTS back on your system. See the FORTRAN-IV installation guide for instructions on how to do this.

INSTALLATION

2.10.1 SYSUPD

The RT-11 distributed SYSLIB contains several routines that are FORTRAN specific. Five of those routines have to be modified for FORTRAN-77: IASIGN, ILUN, SECNDS, GETSTR, and PUTSTR. The object module SYSUPD.OBJ on the PDP-11 FORTRAN-77/RT-11 installation kit contains the FORTRAN-77 version of these routines.

If you do not have FORTRAN-IV on your system, then you should update SYSLIB with the FORTRAN-77 version of these routines by typing

```
UNPROTECT SY:SYSLIB.OBJ<RET>
R LIBR<RET>
*SY:SYSLIB.OBJ[-1]=SY:SYSLIB.OBJ,INP:SYSUPD/U/G<RET>
Global? $OVRH<RET>
Global? <RET>
*^C
PROTECT SY:SYSLIB.OBJ<RET>
```

If you intend to use both FORTRAN-77 and FORTRAN-IV on your system, then you will need a version of the five routines for both compilers. In this case you should merge SYSUPD with the FORTRAN-77 OTS by typing

```
UNPROTECT SY:F77OTS.OBJ
LIBR SY:F77OTS/CREATE SY:F77OTS.OBJ,INP:SYSUPD
PROTECT SY:F77OTS.OBJ
```

2.10.2 Installing OTS in SYSLIB

You can choose to combine the FORTRAN-77 OTS library with RT-11's System Subroutine Library, SYSLIB. The advantage of doing this is that you will not need to specify ",SY:F77OTS" in each LINK command you type. The FORTRAN-77 system subroutines will automatically be linked into your job images. You should not choose this option if other OTS libraries are currently installed in your SYSLIB.

If you want to install the FORTRAN-77 OTS in SYSLIB, you can do so by typing the following sequence of commands:

```
UNPROTECT SY:SYSLIB.OBJ<RET>
R LIBR<RET>
*SY:SYSLIB.OBJ[-1]=SY:F77OTS.OBJ,SYSLIB.OBJ/G<RET>
Global? $OVRH<RET>
Global? <RET>
*^C
PROTECT SY:SYSLIB.OBJ<RET>
```

2.11 RENAME THE COMPILER

RT-11 has a Keyboard Monitor command to run FORTRAN-IV, but none for FORTRAN-77. However, if you rename the FORTRAN-77 compiler to FORTRA.SAV (the FORTRAN-IV compiler name), you will be able to use the FORTRAN command to run the FORTRAN-77 compiler. Most of the FORTRAN switches will work with FORTRAN-77, but since the FORTRAN-77 compiler has more CSI switches than are supported by the Keyboard Monitor, you will have to revert to the CSI method of running the FORTRAN-77 compiler if you want to use certain compiler switches.

If you also intend to use the FORTRAN-IV compiler on your system, you won't be able to rename the FORTRAN-77 compiler to FORTRA.SAV unless you first rename the FORTRAN-IV compiler to something else.

INSTALLATION

If you will use the SJ/FB version of FORTRAN-77, you can rename your FORTRAN-77 compiler as follows:

```
RENAME SY:F77.SAV SY:FORTRA.SAV<RET>
```

If you will use the XM version of FORTRAN-77, you can rename your FORTRAN-77 compiler by typing

```
RENAME SY:F77XM.SAV SY:FORTRA.SAV<RET>
```

Note that you CANNOT rename both the SJ/FB AND XM compilers in this manner.

After renaming the compiler, you must remember to substitute the name "FORTRA" for all references to F77 or F77XM in the documented examples. If an example were to suggest typing

```
R F77
```

you should type

```
R FORTRA
```

2.12 ERRORS

If you had trouble in transferring files from your distribution diskettes, first check to see what types of errors are being reported. The most common errors will reflect a lack of contiguous free disk space on the system disk. If this is the case, SQUEEZE your disk and, if possible, delete unnecessary files or transfer files that you do not need to another disk.

If you find that you cannot transfer data because of media errors on the distribution media, contact your supplier for replacement of the media.

CHAPTER 3

OPTIONAL MODULES

The distribution kit includes a number of optional OTS modules. After creating 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. During the installation procedure described in Chapter 2, the desired optional modules are copied to the system.

3.1 VIRTXM

If you use VIRTUAL arrays in your FORTRAN-77 programs, then you will need the module VIRTXM.OBJ on your system. The PDP-11 FORTRAN-77/RT-11 User's Guide explains how to link this object module with programs that use VIRTUAL arrays. The following example shows how you could link VIRTXM with a program that uses VIRTUAL arrays:

```
LINK MAIN,SY:VIRTXM/XM<RET>
```

3.2 SHORT ERROR TEXT

The ASCII text for FORTRAN-77 OTS is all contained in one module that requires approximately 1000 words of memory. If you need additional memory for your executable job, you can replace the OTS error module with an alternate version, SHORT.OBJ, that requires only one word of memory. With this alternate version, the OTS will generate complete error reports, but will omit the one-line description of the error condition. The PDP-11 FORTRAN-77/RT-11 User's Guide contains a complete list of OTS error numbers and message text. The following example shows how you could link a job with the short error-text module if SHORT.OBJ is on your system device:

```
LINK MAIN,SY:SHORT<RET>
```

3.3 F77MAP

When the FORTRAN-77 compiler encounters an intrinsic function, it translates its FORTRAN-77 name into an internal OTS name. If you wish to refer to the FORTRAN-77 intrinsic functions by their FORTRAN-77 name, then you should link your job with the module F77MAP.OBJ in order to avoid unresolved reference errors during linking. F77MAP.OBJ provides the translation of FORTRAN-77 intrinsic function names to internal OTS names.

For example, F77MAP.OBJ maps the FORTRAN-77 intrinsic function name SIN to the internal OTS name \$SIN by means of the following MACRO code:

OPTIONAL MODULES

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

F77MAP.OBJ contains an object module similar to this for each of the FORTRAN-77 intrinsic functions.

3.4 F77EIS

F77EIS.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-77 programs that do not do floating-point arithmetic to run on a machine that has the extended instruction set but does not have a floating-point processor.

3.5 F77CVF

Object module F77CVF.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.

3.6 F77NER

Object module F77NER.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. F77NER.OBJ reduces job size by about 375 words over the standard module.

3.7 F77RAN

F77RAN.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 F77RAN.OBJ at link time.

CHAPTER 4

OPTIONAL INSTALLATION PROCEDURE

The distribution kit contains an IND command file called `INSTAL.COM` that can assist you in the installation procedure described in Chapter 2. You should read the entire installation procedure in Chapter 2 before attempting this optional procedure. The IND command file does not perform all the installation steps; some must still be performed manually as described in Chapter 2. The installation steps performed by the IND command file are:

- Load and assign the distribution volume (Section 2.4).
- Copy the required files to the system device (Section 2.6).
- Copy the optional OTS modules to the system (Section 2.9).
- Install OTS in SYSLIB (Section 2.10).

4.1 STEPS OF THE OPTIONAL PROCEDURE

To use this optional procedure you should:

- Back up the distribution kit (Section 2.1).
- Decide what part of the distribution you want (Section 2.2).
- Free up available memory space (Section 2.3).
- Make room on the system device (Section 2.5)
- Run the IND command file.
- Patch the linker (Section 2.7).
- Run the verification test (Section 2.8).
- Rename the compiler (Section 2.11)
- Move `VIRTXM` to the system.

4.2 RUNNING THE IND COMMAND FILE

Copy the IND command file from the distribution kit to your system by typing

```
COPY ddn:INSTAL.COM DK:<RET>
```

The expression "`ddn`" represents the name of the device holding the distribution media. If you are familiar with IND, you may want to get

OPTIONAL INSTALLATION PROCEDURE

a listing of INSTAL.COM so you can see what it is actually doing.

If you are performing the installation from magtape or RX01 floppy, the IND command file will transfer all the distribution kit files to the default device DK:, so you must make enough room on DK: to hold the entire distribution. This amounts to approximately 900 blocks.

After you have performed the manual installation steps listed above, run the IND command file by typing

```
R IND<RET>
*INSTAL<RET>
```

The procedure will begin by identifying itself;

```
; FORTRAN-77/RT-11 Installation procedure for xxxxxx
; Version 5.0
```

If the distribution is RX01 or magtape, the files will now be copied to your default device DK:. On RL01 and RX50 distributions, this step is skipped.

```
; This procedure will copy the entire distribution to DK: then build
; the working FORTRAN-77 on SY:
```

If the input device is magtape, the files are copied as follows:

```
.COPY INP:*/POSITION:-1 DK:
```

If the input device is RX01 diskette, the files are copied like this:

```
; Begin by inserting disk number 1 in drive ddn:
* Are you ready? [Y/N] Y
;
.COPY/EXCLUDE INP:INSTAL.COM DK:
;
; Now replace disk 1 with disk 2
* Are you ready? [Y/N] Y
;
.COPY INP:*. * DK:
```

All Distributions: Next, you will be asked to enter the name of the device where the distribution volume is mounted. Remember to include the colon in the device name you specify.

Enter the name of the device holding the distribution files [ddn:]

The procedure will use RT-11's ASSIGN command to assign the logical name INP to the device you have named, and will assign the name OUP to the system device for output.

```
.ASSIGN ddn: INP
.ASSIGN SY: OUP
```

The system device is checked for the presence of SYSLIB.OBJ and LIBR.SAV. These files are necessary for proper installation. If they are not present, you will be told to install them now.

```
; Please locate the distributed RT-11 SYSLIB.OBJ now, and make a copy
; of it on 'OUTDEV'. Then RERUN this command file (INSTAL.COM).
```

or

```
; Please make a copy of LIBR.SAV on your system disk. Then RERUN
; this command file (INSTAL.COM).
```

OPTIONAL INSTALLATION PROCEDURE

Professional 300 series only: If you are installing FORTRAN-77 on a Professional 300 series computer, the procedure inspects the mounted distribution disk to see whether it is the one holding F77XM.SAV. If it is not, then the following message is printed.

```
; PRO-350 Computers must use the XM version of the FORTRAN-77 compiler.
; This version, along with all other required modules are present on
; distribution diskette number 2. Please be sure that this disk is in
; place on DZn: at this time. Then rerun this command file.
```

MICRO PDP-11 Only: If you are installing FORTRAN-77 from an RX50 diskette mounted on a MICRO PDP-11, this message appears:

```
; The RX50 distribution is supplied with a complete FORTRAN-77 system
; on each diskette; disk 1 contains the SJ/FB version of the compiler
; while disk 2 holds the XM version. If the disk that is now loaded
; does not hold the compiler that you want, replace it now with the
; other diskette.
* Is the correct diskette mounted and ready to go? [Y/N] Y
```

If you answer NO, you are told to

```
; Replace the diskette with the other one, and then re-run this command
; file.
```

4.2.1 Steps Common to all Distribution Media

The installation procedure allows you to choose whether the FORTRAN-77 Object Time System Library should be combined with RT-11's System Subroutine Library. The first option keeps these libraries separate, and should be selected if it is desired to keep RT-11 SYSLIB as small as possible. The second option should be chosen if there are other object time system libraries being used on the system. Option 3 is advised if there are no other language compilers installed on your system, and FORTRAN-77 will be your principal programming language. If you are uncertain about which option to choose, refer to section 2.10 for further information on this subject.

```
; Do you want your default system library SYSLIB.OBJ to include
;
; 1) only the RT-11 distributed SYSLIB.OBJ (unchanged)
; 2) SYSLIB plus FORTRAN-IV (FORLIB) (unchanged SYSLIB+FORLIB)
; 3) SYSLIB plus FORTRAN-77 (F77OTS) (new combined SYSLIB+F77OTS)
;
* Enter option number: [0] 3<RET>
```

If you select option 3, the following path is taken:

```
; The original SYSLIB will now be renamed to SYSLIB.DIS (distributed)
; in order that this process can be repeated if necessary. Then, a
; new copy of SYSLIB will be made, and the updates performed.
;
.RENAME OUP:SYSLIB.OBJ OUP:SYSLIB.DIS
.PROTECT OUP:SYSLIB.DIS
```

A temporary KMON command file for the LIBR utility is created and executed. The library commands update the system library (SYSLIB) with FORTRAN-77 specific modules IASIGN, ILUN, SECNDS, GETSTR, and PUTSTR. Then, the FORTRAN-77 Object Time System is combined with the updated SYSLIB.

OPTIONAL INSTALLATION PROCEDURE

```
.$@DK:F77LIB.TMP
.R LIBR
*OUP:SYSLIB.OBJ[-1]=OUP:SYSLIB.DIS,ddn:SYSUPD/U/G
*Global? $OVRH
*Global?
*OUP:SYSLIB.OBJ[-1]=ddn:F77OTS.OBJ,OUP:SYSLIB.OBJ/G
*Global? $OVRH
*Global?
*^C
```

After completion, the temporary command file is deleted.

```
.DELETE/NOQUERY DK:F77LIB.TMP
```

On the other hand, if options 1 or 2 were chosen, the FORTRAN-77 OTS library will not be combined with SYSLIB, but will be copied directly to the system device like this:

```
; The F77 OTS will be copied to SY:. The SYSLIB update modules
; in SYSUPD.OBJ will be merged into the FORTRAN-77 OTS library.
```

Further, only if option 1 was chosen, you are asked whether the F77 OTS should be renamed to "FORLIB.OBJ". This is only recommended in installations that must use the linker's /F switch.

```
; Should F77OTS be given the name FORLIB.OBJ to be compatible with the
* linker's /F switch? [Y/N] NO
;
.LIBR OUP:F77OTS/CREATE DK:F77OTS.OBJ,DK:SYSUPD
```

In all cases, the compiler SAV images and message file are now copied to the system device.

```
; Copying the F77 compiler and message file to the system disk...
;
.COPY DK:(F77*.SAV,F77COM.MSG) OUP:
```

In the final phase, you are asked whether to copy various optional object modules from the distribution volume to your system device. Most of these modules are rarely used, however, you should review the description of these modules in chapter 3. Also note that if you neglect to install any of these modules now, you can retrieve them from the distribution volume when needed at a later date.

RX01 and Magtape: If your default device DK: is the same as your system device SY:, you should answer NO to the first question and skip now to section 4.3, because these files already exist on SY:.

```
; There are several optional object modules supplied in the
; distribution. They are explained in detail in the
; FORTRAN-77/RT-11 Installation Guide.
;
Do you want to copy any of the optional modules to SY:? [Y/N] Y
;
Do you want to copy F77MAP.OBJ? [Y/N] N
;
Do you want to copy F77CVF.OBJ? [Y/N] N
;
Do you want to copy F77EIS.OBJ? [Y/N] Y
.COPY ddn:F77EIS.OBJ OUP:
;
Do you want to copy F77NER.OBJ? [Y/N] N
;
Do you want to copy F77RAN.OBJ? [Y/N] N
```

OPTIONAL INSTALLATION PROCEDURE

```
;
Do you want to copy SHORT.OBJ? [Y/N] Y
.COPY ddn:SHORT.OBJ OUP:
;
; This completes the FORTRAN-77/RT-11 installation.
;
@ <EOF>
```

4.3 RUNNING THE VERIFICATION TEST

After the IND command file installation steps have been performed, and after you have patched the linker (Section 2.7), you are ready to verify the installation (Section 2.8). If you have installed the FORTRAN-77 OTS in SYSLIB, then when you link the test program, as described in section 2.8.2, you only need to type

```
LINK F77TST<RET>
```

If the verification test is successful, then you can go on to rename the compiler (Section 2.11).

4.4 MOVING VIRTXM TO THE SYSTEM

The IND command file procedure does not ask you whether you want to move the optional module VIRTXM.OBJ to the system. If you intend to use virtual arrays in your FORTRAN-77 programs, then you need this module. You should copy it from the distribution kit by typing

```
COPY INP:VIRTXM.OBJ SY:<RET>
```

CHAPTER 5

SYSTEM PERFORMANCE AND TAILORING

This chapter discusses factors affecting compiler performance and describes options you may choose when installing PDP-11 FORTRAN-77/RT-11 on your system. It also describes compiler options you may choose during installation.

5.1 FACTORS AFFECTING COMPILER PERFORMANCE

During execution, the compiler opens a temporary scratch workfile that is used as an overflow area for several internal tables it must build. As each internal table is built, storage is dynamically assigned to it in memory in units of 256 words, called a page. If the dynamic storage memory fills up, then before another page can be allocated, the compiler must move an old page from memory to the disk workfile. And if the compiler must later refer to a page that has been moved to the workfile, it will have to move some other page to the workfile before bringing the needed page back into the dynamic storage memory.

Having to move pages back and forth between memory and the workfile can slow the compiler down significantly; so the compiler will run most quickly if it can fit the internal tables it builds completely into dynamic storage memory.

Tests indicate that for average-size programs compiler performance continues to increase significantly as the size of the dynamic storage area is increased from the minimum allowable (2 pages) to about 14 pages in size, and that little improvement is obtained as it is increased beyond 14 pages. The XM version of the compiler always uses a fixed size dynamic storage area of 14 pages, but the size of the dynamic storage area for the SJ/FB version of the compiler will depend on the amount of background memory available at run time.

For the SJ/FB version of the compiler the size of the dynamic storage area is affected by the number and size of device handlers, presence of the USR in a NOSWAP condition, and the presence of SYSTEM or FOREGROUND jobs such as the queue manager, single line editor, spooler, or the VTCOM communications package. The size of dynamic storage area can be increased by unloading device handlers that are not needed for the compilation and by aborting and unloading as many system/foreground jobs as possible. As a general rule, the single line editor should be OFF and the USR should be set to SWAP.

The size of the dynamic storage area in pages can be determined by inspection of a compiler listing made with the /A switch. This switch causes the compiler to write a summary of workfile and dynamic workspace usage to the end of each source module listing. Use the RT-11 Keyboard Command SHOW MEMORY to see the amount of space available to the background process. Consult your RT-11 System User's Guide for complete information on how to unload device drivers and how to use the SHOW MEMORY command.

SYSTEM PERFORMANCE AND TAILORING

Larger programs cause the compiler to generate larger internal tables, so they have a greater chance of degrading performance. If your program has a very long main section or very long subroutines, you may want to consider breaking the code into smaller modules.

When you cannot avoid a situation where the compiler is moving lots of pages to and from the disk workfile, you may want to consider forcing the compiler to put the workfile on a faster access device, such as VM:. When the compiler opens the workfile, it looks for a logical device name WF:. If the name WF: is assigned to a physical device, it opens the workfile there. If the name WF: is not assigned, then it opens the workfile on the default device DK:. To force the workfile to a specific device, simply assign the logical name WF: to it. For example, if you want to force the workfile to be created on VM:, then you should type

```
ASSIGN VM WF<RET>
```

If there is not adequate space on the logical device WF: to open the workfile (the default size is 128 blocks), the compiler will abort the error message

```
F77 -- FATAL 01 * Open error on work file
```

Before you can run the compiler, you must make enough free memory available. This can be done by temporarily stopping system jobs and removing unnecessary handlers, as described in section 2.3. The XM compiler requires 12.5K words of kernel memory and 58K words of extended memory. The SJ/FB compiler requires a minimum of 22K words of memory, but it will perform significantly better if it has an additional 1K word of memory available.

5.2 COMPILER TAILORING

The distributed version of the compiler may have options that do not exactly match your needs. For example, you may prefer to have an 80 column listing instead of a 132 column listing or you may desire integers to be interpreted as INTEGER*4 by default instead of INTEGER*2. Table 5-1 lists the user-modifiable features of the compiler.

NOTE

Modification of a switch's default setting may affect compiler operation under the FORTRAN Keyboard Command. For this reason, changes in compiler switches are not recommended.

SYSTEM PERFORMANCE AND TAILORING

Table 5-1
User Changable Flags

Affected flag	Flag Action
/A	Compilation Statistics
/B	Debugger Flag
/C	Maximum number of continuation lines
/D	Debug line inclusion
/E	Extend Flag
/F	Workfile length
/I	Array checking
/L	Default listing value
/O	Optimizer switch
/Q	Line Printer Width
/S	Default Trace code
/T	INTEGER*4 switch
/W	Warning message display
/Y	Default Standard checking
/Z	Pure code & data

To modify the compiler image file, F77.SAV or F77XM.SAV, use the Save Image Patch Program (SIPP). This program is described in the RT-11 System Utilities Manual.

In order to use SIPP, you need to know where within the image (or .SAV) file, the modifications are to be made. Appendix A presents the assembly code for two PSECTs that contain user-modifiable variables. The PSECTs are named SWITCH and WRKNAM.

5.2.1 PSECT SWITCH

PSECT SWITCH contains the default settings for the compiler. Only the switches itemized in Table 5-1 should be modified.

5.2.2 PSECT WRKNAM

PSECT WRKNAM contains the default device and file name of the workfile (in RADIX-50) that the compiler uses to hold overflow from the internal tables that it builds in memory. When the compiler attempts to open the workfile, it first uses the logical device name WF:. You can use the RT-11 ASSIGN command to assign WF to any supported device on your system. If WF: is not assigned to a physical device, the open fails and the compiler then attempts to open the workfile on the device named in PSECT WRKNAM. The standard device name supplied in PSECT WRKNAM is DK:. You may find it convenient to change the device name to some more appropriate device for your system.

5.2.3 Modifying the compiler image

If you want to modify the default values of the switches in PSECT SWITCH or change the workfile device in PSECT WRKNAM, you can use the Save Image Patch Program (SIPP) to modify the memory image file of the compiler. For complete information on the SIPP commands, see Chapter 20 of the RT-11 System Utilities Manual. To do this you need to know where the PSECTs start in the compiler image. Table 5-2 presents the required addresses for the two different distributed images.

SYSTEM PERFORMANCE AND TAILORING

Table 5-2
PSECT addresses (OCTAL)

PSECT	F77.SAV	F77XM.SAV
SWITCH	024032	023436
WRKNAM	024104	023510

Using these addresses, and information from APPENDIX A, you can patch the binary image.

If you have a small system, you may not have SIPP on your normal system disk. If you do not, copy SIPP from your distribution backup of RT-11 and onto a working medium. Use this copy of SIPP to patch the compiler image.

If the compiler you want to modify is protected, unprotect it by typing

```
UNPROTECT SY:F77.SAV<RET>
```

or

```
UNPROTECT SY:F77XM.SAV<RET>
```

To call SIPP you can enter

```
.R SIPP<RET>          if the program is on your system disk
```

or

```
.RUN SIPP<RET>       if the program is on your default device.
```

SIPP will respond with the * prompt from the Command String Interpreter. To this respond

```
SY:F77.SAV<RET>
```

or

```
SY:F77XM.SAV<RET>
```

depending on which compiler you wish to modify.

After you have entered the initial command string, SIPP will print a series of prompts on your terminal. The first prompt will ask for the segment number of the compiler. Since you are working in the root you will answer 0 as shown:

```
*Segment? 0<RET>
```

The next prompt is for the base address within the program where the modifications will be made. This will correspond to the PSECT address specified in Table 5-2. Suppose, for illustration, that you wish to modify the default size of the F77XM.SAV compiler workfile size to be the maximum size, which is 256 decimal (400 octal) blocks. Then your response will look like this:

```
*Base? 023436<RET>
```

After the base is specified, SIPP prompts you to enter the offset from this base to the word or byte you want to modify. Refer to APPENDIX A to find this information. In the .PSECT SWITCH you will see the global name WFOOO:: with a value of 128. decimal (200 octal): this is the location you must modify. Note that the offset for this parameter is 12 (octal); this is the value you will enter in response to the prompt.

```
*Offset? 12<RET>
```

SYSTEM PERFORMANCE AND TAILORING

SIPP now prints a header and completes the information that you have specified:

Segment	Base	Offset	Old	New?
000000	023436	000012	000200	

Under the column marked Old, SIPP prints the contents of the currently open location. Under the column designated New?, you can enter either a new value for the current location and/or a command. In this case 400<RET> should be entered. Now enter CTRL/Y. CTRL/Y causes SIPP to complete the current patching session by installing the patch, closing the modified file, and prompting you with another *. A CTRL/C will allow you to exit SIPP with the compiler modifications made.

CHAPTER 6

RELEASE NOTES

6.1 VIRTUAL ARRAYS NOT SUPPORTED UNDER FB AND SJ MONITORS

This distribution of PDP-11 FORTRAN-77/RT-11 does not provide support for VIRTUAL arrays under the FB or SJ monitors. If you need to use VIRTUAL arrays, you must run under the XM monitor.

6.2 USING AN UNPATCHED LINKER

Section 2.8 describes a patch that should be applied to the linker. If you are using a linker that does not have the patch applied, you may have difficulty linking with certain object modules produced by the FORTRAN-77 compiler. For these modules the linker will produce an "Invalid GSD" error message and abort. The problem is related to the linker's management of memory space. If you encounter this problem, and you do not have access to a patched linker, then there are some things you can try that may cause the linker to accept the troublesome object modules.

If more background memory is available to the linker, it tends to be less sensitive to the problem. Make more background memory available by unloading unnecessary handlers, stopping foreground and system jobs, and turning off the Single Line Editor and the spooler. If you are using the SJ or FB monitor, set the USR to SWAP.

The linker will sometimes accept an offending module if it is recompiled with different switch options. Try using different combinations of the /S (trace), /O (optimization), and /B (debug) switches.

6.3 UNFORMATTED SEQUENTIAL ARRAYS

Unformatted sequential arrays of type LOGICAL*1 or BYTE are transferred to memory in a byte for byte format rather than word for byte. This makes more efficient use of disk space, but is not consistent with the default form used by the FORTRAN-IV compiler.

6.4 OTSHK1

When a FORTRAN-77 program starts execution, the first thing it does is call the OTS initialization routine \$OTI. Just before the return from \$OTI, the module OTSHK1 is called. In the distributed version of OTS, OTSHK1 sets the lowercase bit in the Job Status Word to allow lower case I/O to the terminal.

If you would like to perform different initialization procedures at

RELEASE NOTES

this point, replace the OTSHK1 with your own version.

6.5 NON-FILE STRUCTURED LOOKUPS

In contrast with the FORTRAN-IV implementation, opening a device in a non-file structured manner using the OPEN or ASSIGN statements is not possible. This restriction prevents users from accidentally destroying directories.

6.6 MIXING COMMON DATA TYPES

PDP-11 FORTRAN-77/RT-11 does not allow character data and numeric data in the same COMMON block.

6.7 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 for the other intrinsic functions, on the basis of the data type of arguments in the function reference.

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

RELEASE NOTES

6.8 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.

6.9 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 ABC in the subroutine BBBB parameter list.

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

6.10 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
```

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

6.11 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:

RELEASE NOTES

```
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
```

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

APPENDIX A

COMPILER PSECTS THAT CAN BE MODIFIED

This Appendix contains the definitions of the PSECTS described in Chapter 5. Table A-1 shows the PSECT offset addresses for the SJ/FB and XM compilers.

Psect SWITCH

```

.PSECT SWITCH,D

0      SF000:: 0      ; /A Statistics flag: 0=no, 1=yes
2      DB000:: 0      ; /B Debugger flag
4      CO000:: 19.    ; /C Maximum number of continuation lines
6      DE000:: 0      ; /D Debug lines: 0=omit, 1=include
10     EX000:: 0      ; /E Extend flag
12     WF000:: 128.   ; /F Workfile length in blocks
14     CG000:: 0      ; /G Code generator debug switch
16     FI000:: 0      ; /H Final debugging switch
20     CK000:: 0      ; /I Array checking: 0=off, 1=on
22     LI000:: 2      ; /L Listing option value
24     OP000:: 3      ; /M Optimizer debugging switch
26     OS000:: 0      ; /O Optimize switch
30     SP000:: 1      ; /P Spool list file: 0=no, 1=yes
32     LP000:: 0      ; /Q Printer width: 0=80, 1=132 columns
34     TR000:: 3      ; /S Trace code: 0=NON,1=NAM,2=BLO,7=ALL
36     I4000:: 0      ; /T INTEGER4: 0=no, 1=yes
40     WR000:: 0      ; /W Warnings: 0=omit, 1=include
42     F7700:: 1      ; /X F77 CODE: 0=no, 1=yes
44     ST000:: 0      ; /Y Standard variable
46     RO000:: 0      ; /Z Readonly pure code/data: 0=off,1=on
50     SUP00:: 1      ; Superceding of output files: 1=yes,0=no
    
```

Psect WRKNAM

```

.PSECT WRKNAM,D

WRKNAM::.RAD50 /DK F77A TMP/
    
```

Table A-1
PSECT addresses (OCTAL)

PSECT	F77.SAV	F77XM.SAV
SWITCH	024032	023436
WRKNAM	024104	023510

INDEX

- Associated documents, iii
- Compiler renaming, 2-10
- Compiler switches, 5-1 to 5-3
- Compiler tailoring, 5-2
- Conventions, iv
- DCL, 2-10
- Distribution files, 1-2
- Distribution media
 - magtape, 2-5, 4-2
 - RL01, 2-4
 - RX01, 2-4 to 2-5, 4-2
 - RX50, 2-4 to 2-5
- Errors
 - compiler, 2-7 to 2-8, 5-2, 6-3
 - installation, 2-11
 - linker, 2-8, 6-1
 - OTS, 2-8, 3-1 to 3-2
 - workfile, 5-2
- F77.SAV, 1-2, 2-1, 2-4 to 2-6, 5-3
- F77COM.MSG, 1-2, 2-4 to 2-6, 2-8
- F77CVF.OBJ, 1-2, 3-2
- F77EIS, 3-2
- F77EIS.OBJ, 1-2
- F77MAP.OBJ, 1-2, 3-1
- F77NER.OBJ, 1-2, 3-2
- F77OTS.OBJ, 1-2, 2-4 to 2-6, 2-8 to 2-9
- F77RAN.OBJ, 1-2, 3-2
- F77TST.FOR, 1-2, 2-7 to 2-8, 4-5
- F77XM.SAV, 1-2, 2-1, 2-4 to 2-6, 5-3
- Floating-point processor, 2-8, 3-2
- Formatted output
 - optional module, 1-2
- FORTTRAN-IV, 2-9 to 2-10
- GETSTR, 2-10
- IASIGN, 2-10
- ILUN, 2-10
- INP:, 2-5
- INSTAL.COM, 1-2, 2-1, 2-4, 4-1
- Installation, 1-2
- Intrinsic functions, 3-1
- Invalid GSD, 2-6, 6-1
- Linker, patching, 2-6, 6-1
- Mixing COMMON data types, 6-2
- Monitor
 - FB, 2-1, 2-4, 2-7, 6-1
 - SJ, 2-1, 2-4, 2-7, 6-1
 - XM, 2-1 to 2-2, 2-4, 2-7
- Non-file structured lookups, 6-2
- Optional modules, 2-2, 2-4, 2-9, 3-1
- OTS
 - EIS replacement module, 1-2
 - error text file, 1-2
 - error-reporting module, 1-2
 - Errors, 3-1
 - installing in SYSLIB, 2-10
 - OTSHK1, 6-1
- PDF name-mapping module, 1-2
- Performance, 5-1
- Professional 300 series, 1-1, 2-2, 2-4 to 2-5
- PSECTS
 - locations, 5-4, A-2
 - SWITCH, 5-3
 - WRKNAM, 5-3
- PUTSTR, 2-10
- Randon-number generator, 1-2
- Requirements, 1-1
 - floating-point, 1-1
 - floating-point processor, 2-8
 - memory management, 1-1
 - memory size, 1-1
 - RT-11, 1-1
 - system device space, 2-5
 - verification, 1-1
- SECNDS, 2-10
- SHORT.OBJ, 1-2, 3-1
- SHOW MEMORY, 5-1
- SIPP, 2-6, 5-3
- SYSLIB, 2-9
 - update, 1-2, 2-9 to 2-10
- System tailoring, 5-1
- SYSUPD.OBJ, 1-2, 2-10
- Verification procedure, 2-7, 4-5
- Virtual arrays, 1-2, 3-1, 6-1
- VIRTXM.OBJ, 1-2, 2-9, 3-1, 4-5
- WF:, 5-2
- Workfile, 5-1 to 5-2