
FORTRAN VII COMPILER

Release Notes

04-459 M95



Concurrent 
Computer Corporation

CONTENTS

DISCLAIMER	ii
PREFACE	iii
DOCUMENT ORGANIZATION	iv
1. PRODUCT OVERVIEW	1
2. PRODUCT IDENTIFICATION, AVAILABLE MEDIA / DOCUMENTATION	2
2.1 UNIVERSAL FORTRAN LANGUAGE SYSTEM - package 04-149	2
2.2 SERIES 3200 FORTRAN ENHANCEMENT PACKAGE - package 04- 175	3
2.3 FORTRAN VII PERFORMANCE UTILITIES - package 04-177	3
2.4 FORTRAN VII UNIVERSAL COMPILER UPGRADE - package 04-180	4
2.5 FORTRAN VII COMPILER - package 04-459	4
3. PACKAGE DEFINITION	5
3.1 Files	5
3.2 Documentation	7
4. SPECIFIC REQUIREMENTS	8
4.1 Software	8
4.2 Hardware	8
5. UNPACKAGING AND INSTALLATION	9
5.1 Magnetic Tape (1600 and 1/4" Streamer Tape)	9
6. POST-INSTALLATION PROCEDURES	10
7. BUILDING THE COMPILERS AND RTL	11
7.1 OPTIMIZING AND DEVELOPMENT COMPILERS	11
7.2 BUILDING RUN-TIME LIBRARY (RTL)	13
7.3 THE FORTRAN VII EASE-OF-USE PROCEDURES	15
7.4 HIGH LEVEL OPERATOR COMMAND LEVEL (HLOC)	17
7.5 ESTABLISHING THE WRITABLE CONTROL STORE (WCS) SUPPORT PROGRAMS	19
7.6 WRITABLE CONTROL STORE (WCS) RUN-TIME LIBRARIES (RTLs)	20
8. MICROCODE RUN-TIME LIBRARY (RTL) ROUTINES	21
9. FORTRAN VII Performance Utilities	21
APPENDIX A - RELEASE NOTES	22
APPENDIX C - PROBLEM REPORT PROCEDURES	30

DISCLAIMER

The information in this document is subject to change without notice and should not be construed as a commitment by the Concurrent Computer Corporation. The Concurrent Computer 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 it can be used or copied only in a manner permitted by that license. Any copy of the described software must include the Concurrent Computer Corporation copyright notice. Title to and ownership of the described software and any copies thereof shall remain in the Concurrent Computer Corporation.

The Concurrent Computer Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Concurrent Computer Corporation.

The Concurrent Computer Corporation,
2 Crescent Place, Oceanport, N.J. 07757

© 1990 by The Concurrent Computer Corporation

Printed in the United States of America

PREFACE

The **Software Release Notes** are designed to supplement the information in this product's user guides and reference manuals. This guide does not eliminate the need to read and understand the information contained in the product documentation package, as it is meant solely to be an aid to the user in performing initial installation of the product and to provide product overview information.

DOCUMENT ORGANIZATION

This packaging information document is applicable to all FORTRAN VII related products. All of the information in this document is applicable to all packages unless otherwise specified.

The various packages related to the FORTRAN product are listed in the following table and described in this document:

Sales Order Number	Package Number	Package Description
S7x-225	04-149	Universal FORTRAN Language System
S70-060	04-175	Series 3200 FORTRAN Enhancement Package
S70-228	04-177	FORTTRAN Performance Utilities Package
S7x-232	04-180	FORTTRAN VII Universal Compiler Upgrade
S7x-224	04-459	FORTTRAN VII Compiler

The cover page of this document identifies the package that has been delivered to you. Please follow the installation steps specific to this package.

1. PRODUCT OVERVIEW

The FORTRAN R06-00 release is a major release of the FORTRAN Language System. Following is a summary of enhancements:

- New edit descriptors \ and \$ to suppress carriage return at the end of an output operation.
- The # substring length specifier.
- New relational operators; Exclusive OR (.EOR.) and NOT (^).
- DO/END DO statements
- DO WHILE statement
- DO UNTIL statement
- New data types: INTEGER*1, LOGICAL*2, BYTE
- Hexadecimal REAL and DOUBLE PRECISION constants.
- O and Z typeless constants.
- MIL STD 1753 bit manipulation routines; ISHFTC, IBSET, IBITS, MVBITS and IBCLR.
- New and Enhanced Compiler Start/Instream options:

Start/Instream Option	Description
INT2	All integer variables with non-explicit lengths treated as INTEGER*2
IBYTE	The BYTE statement will be treated as an INTEGER*1 statement
LBYTE	The BYTE statement will be treated as a LOGICAL*1 statement
LTORBIT	Bit positions counted from left to right (i.e. MSB = Bit 0)
RTOLBIT	Bit positions counted from right to left (i.e. MSB = Bit 31)
TABLES	Dump tables for E/SP
TARGET={3283,3285,3288}	Target code to the MICRO III, MICRO V, and 3280E respectively
XFORT	Source contains special E/SP-generated, non-FORTRAN constructs
(N)F66DO	All DO-loops will be executed at least <i>once</i>
(N)PASSBYADDRESS	All scalar arguments to a subprogram assumed "pass by address"
(N)REENTRANT	Generate reentrant code
(N)TRANSCENDENTAL	Generate RTL calls instead of 328X transcendental micro-instructions
(N)DEBUG	Generate code which includes debugger information
(N)SAFE	The embedded assembly code can be "safely" parallelized by E/SP
(N)OVERLAP	Allow non-standard conforming aliasing of dummy arguments for E/SP
(N)OBJ	Controls generation of object code for E/SP
(N)OPT	Modified to replace F7D with F7O/Z and support for E/SP

- Support for E/SP.
- Expanded compiler tables to increase the compiler's capacity for processing data statements and doing optimization.

2. PRODUCT IDENTIFICATION, AVAILABLE MEDIA / DOCUMENTATION

This section lists the available packages for each of the constituents of the FORTRAN product.

2.1 UNIVERSAL FORTRAN LANGUAGE SYSTEM - package 04-149

Product Title:	Universal FORTRAN Language System Software Package
Part Number:	04-149 R05-00
Sales Order Number:	S7x-226

The Universal FORTRAN Language System Software Packages that are currently available are:

Sales Order Number	Part Number	Software Package
S7x-225-ABC**	04-149M71R05	Universal FORTRAN Language System Software (9-track, 1600 BPI Magnetic Tape) and Documentation Package
S7x-225-ABW**	04-149ML1R05	Universal FORTRAN Language System Software (1/4" Streaming Tape) and Documentation Package
S7x-225-BCM**	04-149M99R05	Universal FORTRAN Language System Documentation Package

** S7x will be one of the four license categories.

2.2 SERIES 3200 FORTRAN ENHANCEMENT PACKAGE - package 04-175

Product Title:	Series 3200 FORTRAN Enhancement Software Package
Part Number:	04-175 R03
Sales Order Number:	S70-060

The Series 3200 FORTRAN Enhancement Software Packages that are currently available are:

Sales Order Number	Part Number	Software Package
S70-060-ABC**	04-175M71R03	Series 3200 FORTRAN Enhancement Software (9-track, 1600 BPI Magnetic Tape) and Documentation Package
S70-060-ABW**	04-175ML1R03	Series 3200 FORTRAN Enhancement Software (1/4" Streaming Tape) and Documentation Package
S70-060-BCM**	04-175M99R03	Series 3200 FORTRAN Enhancement Documentation Package

2.3 FORTRAN VII PERFORMANCE UTILITIES - package 04-177

Product Title:	FORTRAN Performance Utilities Software Package
Part Number:	04-177 R04
Sales Order Number:	S70-228

The FORTRAN Performance Utilities Software Packages that are currently available are:

Sales Order Number	Part Number	Software Package
S70-228-ABC**	04-177M71R04	FORTRAN Performance Utilities Software (9-track, 1600 BPI Magnetic Tape) and Documentation Package
S70-228-ABW**	04-177ML1R04	FORTRAN Performance Utilities Software (1/4" Streaming Tape) and Documentation Package
S70-228-BCM**	04-177M99R04	FORTRAN Performance Utilities Documentation Package

2.4 FORTRAN VII UNIVERSAL COMPILER UPGRADE - package 04-180

Product Title:	FORTRAN VII Universal Compiler Upgrade Software Package
Part Number:	04-180 R04
Sales Order Number:	S7x-232

The FORTRAN VII Universal Compiler Upgrade Packages that are currently available are:

Sales Order Number	Part Number	Software Package
S7x-224-ABC**	04-180M71R04	FORTRAN VII Universal Compiler Upgrade Software (9-track, 1600 BPI Magnetic Tape) and Documentation Package
S7x-224-ABW**	04-180ML1R04	FORTRAN VII Universal Compiler Upgrade Software (1/4" Streaming Tape) and Documentation Package
S7x-224-BCM**	04-180M99R04	FORTRAN VII Universal Compiler Upgrade Documentation Package
** S7x will be one of the four license categories.		

2.5 FORTRAN VII COMPILER - package 04-459

Product Title:	FORTRAN VII Compiler Software Package
Part Number:	04-459 R00
Sales Order Number:	S7x-224

The FORTRAN VII Compiler Packages that are currently available are:

Sales Order Number	Part Number	Software Package
S7x-232-ABC**	04-180M71R00	FORTRAN VII Compiler Software (9-track, 1600 BPI Magnetic Tape) and Documentation Package
S7x-232-ABW**	04-180ML1R00	FORTRAN VII Compiler Software (1/4" Streaming Tape) and Documentation Package
S7x-232-BCM**	04-180M99R00	FORTRAN VII Compiler Documentation Package
** S7x will be one of the four license categories.		

3. PACKAGE DEFINITION

3.1 Files

This table lists the files that belong to the various FORTRAN packages. Note that the naming conventions for some of the files have changed since the previous release. The extensions of Run Time Library (RTL) files have been changed from OBJ to LIB. Also, the math RTL file names have changed from PEMxxx to F7Mxxx.

File Name	Part Number	Description	PACKAGES 04-XXX					
			149	175	176	177	180	459
INFORM.CSS		Restricted Rights Disclosure	yes	yes	yes	yes	yes	yes
F7Z60.TSK	03-376 R06-00	FORTTRAN VII Z compiler Task for MAT Machines	yes				yes	
F7ZMAC60.TSK	03-679 R06-00	FORTTRAN VII Z compiler Task for MAC Machines	yes				yes	
F7O60.TSK	03-207 R06-00	FORTTRAN VII O compiler Task file	yes					yes
F7ZO60.ERR	03-306 R06-00	FORTTRAN VII Z & O compiler Error file	yes				yes	yes
F7LIB60.LIB	07-203F01 R06-00	RTL with no argument checking	yes					yes
F7LIB60A.LIB	07-203F02 R06-00	RTL with argument checking	yes					yes
F7RTL.LIB.CSS	07-203F03 R06-00	RTL Building CSS	yes					yes
R05CMPT.LIB	07-203F04 R06-00	R05 compatible RTL	yes					yes
R05CMPTA.LIB	07-203F05 R06-00	R05 compatible RTL	yes					yes
F7REL60.LIB	07-203F06 R06-00	RELIANCE compatibility RTL	yes					yes
F7M60.LIB	07-305F01 R06-00	System Math Library without argument checking	yes					yes
F7M60A.LIB	07-305F02 R06-00	System Math Library with argument checking	yes					yes
F7M60X.LIB	07-305F03 R06-00	System Math Library WCS version without argument checking(3200 series)	yes	yes				
F7M60AX.LIB	07-305F04 R06-00	System Math Library WCS version with argument checking(3200 series)	yes	yes				
F7M60W.LIB	07-305F05 R06-00	System Math Library WCS version without argument checking(8/32)	yes	yes				

File Name	Part Number	Description	PACKAGES 04-XXX					
			149	175	176	177	180	459
F7M80AW.LIB	07-305F06 R06-00	System Math Library WCS version with argument checking(8/32)	yes	yes				
MICRTL45.WCS	03-480F01 R00-01	Models 3240, 3250, 3250XP and 3200MPS/3260MPS CPU microcode RTL routines	yes	yes				
MICRTL30.WCS	03-480F02 R00-01	Models 3212, 3230, 3230XP, 3230MPS(CPU & APU) microcode RTL routines	yes	yes				
MICRTLAP.WCS	03-480F03 R00-01	3200MPS/3260MPS APU microcode RTL routines	yes	yes				
LDWCS.OBJ	03-170 R01-02	Model 8/32 Writable Control Store(WCS) Load Program	yes	yes				
F6RTLW.OBJ	07-178 R00-00	FORTTRAN VI Writable Control Store(WCS) Run-Time Library	yes	yes				
F7RTL60.ERR	03-223 R06-00	FORTTRAN RTL Error Message File	yes					yes
NEOUF7.TSK	03-445 R00-00	FORTTRAN VII EOU Task (OS/32 R05-02 thru R06-01 Version)	yes					yes
EOU61F7.CSS	03-446 R06-00	FORTTRAN VII EOU CSS (OS/32 R05-02 thru R06-01 Version)	yes					yes
EOUF7.CSS	03-446F01 R06-00	FORTTRAN VII EOU CSS (OS/32 R06-02 Version and above)	yes					yes
F7HLOC.CSS	03-208 R06-00	An ASCII file which builds the High Level FVII compiler CSSs	yes					yes
XPATAB.TSK	03-439F01 R06-00	XPA task image	yes			yes		
XPATAB.CSS	03-439F02 R06-00	CSS to run XPATAB	yes			yes		
CRATAB.TSK	03-439F03 R06-00	CRA task image	yes			yes		
CRATAB.CSS	03-439F04 R06-00	CSS to run CRATAB	yes			yes		
F7PFUT.OBJ	03-439F05 R06-00	Performance Utilities Object Library	yes			yes		

3.2 Documentation

The FORTRAN Language System Documentation Packages consist of the following manuals:

Publication Number	Revision Level	Publication Name	PACKAGES 04-XXX						
			149	175	176	177	180	459	
48-017F00	R04-00	FORTRAN VII Language and Syntax - A Reference	yes						yes
48-010F00	R04-00	Developing Programs with FORTRAN VII - A Guide	yes						yes
48-025F00	R03-00	System Math Library Reference Manual	yes						yes
48-152F00	R03-00	OS System Support RTL Reference Manual	yes						yes
48-095F00	R02-00	Series 3200 FORTRAN Enhancement Package Reference Manual	yes	yes					
04-149M95	R05-00	Universal FORTRAN Language Release Notes	yes						
04-175M95	R03-00	Series 3200 FORTRAN Enhancement Release Notes		yes					
04-177M95	R04-00	FORTRAN Performance Utilities Release Notes				yes			
04-180M95	R04-00	FORTRAN VII Universal Compiler Upgrade Release Notes					yes		
04-459M95	R00	FORTRAN VII Compiler Release Notes							yes

4. SPECIFIC REQUIREMENTS

4.1 Software

The software packages listed below must be of proper revisions to use all the FORTRAN Language System packages. The package number and minimum revision level of each is as indicated:

Product Name	Sales Order Number	Part Number
OS/32 (R08-02)	S7x-016	04-082-R17
MTM (R08-02)	S7x-017	04-083-R11
LINK (R08-02)	S80-016	03-242

NOTE

CAL code generated by FORTRAN VII compilers cannot be compiled using revisions below CAL/32 R01.

4.2 Hardware

The FORTRAN Language System Packages run on all Concurrent Computer Corporation 32 bit processors.

5. UNPACKAGING AND INSTALLATION

5.1 Magnetic Tape (1600 and 1/4" Streamer Tape)

The files are supplied on magnetic tape in OS/32 Backup format. The unpackaging procedure involves restoring the files from tape to a disk utilizing OS/32 Backup. Please note that the files should be restored onto the system account on the system volume for general purpose use.

After selecting a disk onto which the files are to be copied, review the disk filenames to ensure that they do not conflict with the filenames in this package, as listed in Section 3.1 of this document. Rename any conflicting filenames to some other appropriate name prior to backing down the distribution tape.

NOTE

The files supplied in this package must not be renamed, because the command substitution system (CSS) files require specific filenames as listed in the Package Description section.

To unpackage the files from the system console or Multi-Terminal Monitor (MTM) terminal, mount the distributed tape and run OS/32 BACKUP via the following commands:

```
LOAD BACKUP
```

```
TASK BACKUP
```

```
START ,IN=dev1:,OUT=dev2:,LIST=dev3:,VERIFY,CLOSE=REW
```

where,

dev1 is the device name for the magnetic tape drive

dev2 is the device name for the destination disk

dev3 is the device name for the printer or list device

Note that all device names have the standard OS/32 format. They are dependent on the user's system configuration and may be determined by entering the OS/32 command 'DISPLAY DEVICES' at the system console or user's terminal.

NOTE

In running OS/32 BACKUP from an MTM terminal, the user must sign on to an account which has bare disk access and the task account privilege enabled.

For further information on Backup, see the OS/32 System Support Utilities Reference Manual, Publication Number 48-031.

6. POST-INSTALLATION PROCEDURES

Before using the installed software, it is recommended that the following procedures be executed:

- Using OS/32 BACKUP, copy the software files for archiving.
- After marking the disk on and setting the appropriate volume, enter the command substitution system (CSS) call:

INFORM [list-device:]

Licensing information regarding this product will be displayed on the user's terminal by default, or to an optional list-device specified in the CSS call.

7. BUILDING THE COMPILERS AND RTL

The distributed package does not contain object files to build the compilers. The task files are already supplied. However, the Run-Time Library (RTL) must be built before using any of the compilers.

7.1 OPTIMIZING AND DEVELOPMENT COMPILERS

This section is applicable only to the following packages:

- 04-149 -- Universal Fortran Language System
- 04-180 -- Fortran VII Universal Compiler Upgrade
- 04-459 -- Fortran VII Compiler

7.1.1 FORTRAN VII O & Z DEFAULT OPTIONS

The FORTRAN VII O & Z compilers have been built with the following default task options:

NXSVC1	NVFC	NAPCONTROL	UTASK
FLOAT	NRESIDENT	NCONTROL	NCOMMUNICATE
SVCPAUSE	DFLOAT	ROLL	ACCOUNTING
ACPRIVILEGE	NDISC	NUNIVERSAL	NKEYCHECK
NSEGMENTED	NAPAUSE	NINTERCEPT	OVERLAY-IN-TASK
NDTABLES	NAPUONLY	NAPMAPPING	

and the following default LINK options:

TEQSAV=ALL	LU=16	SYSSPACE=XFFFFFF	WORK=(X5000,XAFFFFFF)*
IOBLOCKS=4	PRIORITY=(128,128)	TSW=X3F001000	ABSOLUTE=X100

* F7O and F7ZMAC compiler tasks are linked with: WORK=(X5000,X90000)

The number of logical units available to the compiler for purposes of source inclusion(via \$INCLUDE) is seven(7); that is from LU 9 to LU 15. This represents an LU option specification of 16.

For more information concerning LINK and its options, consult OS/32 LINK Reference Manual, Publication number 48-005.

WARNING

If Writable Control Store (WCS) is loaded on a 3200 series processor, with the exception of 3280, the FORTRAN VII Z and O compilers will assume the presence of the FORTRAN Enhancement Package (FEP) and automatically switch to the use of the microcoded enhancement routines. This can cause a system crash if the FEP is not loaded. To run these compilers on a system having a WCS loaded with user-written microcode routines, apply the following patch using OS/32 PATCH utility:

```
>IMAGE F7x.TSK,F7NWCS.TSK
>GET *TASK
>EXAMINE 124,2
```

```
124:0000 0001
>MODIFY 124,0,0
>EXAMINE 124,2
124:0000 0000
>SAVE
>END
```

The new task is to be used with the distributed CSSs. It must either be renamed to the oldname or the CSSs must be changed to load this new task.

For those installations which wish to change some of the task options, OS/32 PATCH Utility can be used. Refer to **OS/32 PATCH User Guide**, Publication Number 48-016, for more information.

7.2 BUILDING RUN-TIME LIBRARY (RTL)

This section is applicable only to the following packages:

04-149 -- Universal Fortran Language System
04-459 -- Fortran VII Compiler

Concurrent Computer Corporation has divided the FORTRAN VII R06-00 Run-Time Library into three parts so that other products may take advantage of the separate functions incorporated in each part.

- The first, called the **FORTRAN VII Run-Time Library (07-203)**, contains all the environmental support routines necessary for a FORTRAN task, such as the Input/Output (I/O) Formatter, Language Extensions, Real-Time functions, MPS support routines and so on.
- The second, called the **System Math Library (07-305)**, contains all the mathematical routines, such as SIN, COS, and so on.
- The third, called the **Fortran VII R05 compatible Run-Time Library** provides object compatibility so that object generated by R05 compilers can be linked with object generated by R06 compilers.

Additionally, there are more files for math libraries: two files for support of WCS for 3200 series and two files for support of WCS for 8/32 processors. Note that a package may not contain all six variants of the math library.

All Libraries have been generated with and without argument checking; thus there are two versions of each library. (See Section 3.1 of this document for a complete part description.)

More information on the WCS libraries may be found in the **System Mathematical Run-Time Library Reference Manual**, Publication number 48-025, and **Series 3200 FORTRAN Enhancement Package Reference Manual**, Publication Number 48-095.

It is recommended that when a program is being debugged, the user's task should be established using the argument checking run-time library. This will enable the user to find any illegal argument types which are being passed to any library routine.

The system math libraries should first be merged with the FORTRAN VII Run-Time Library so that a complete FORTRAN Run-Time Library is available before using the compiler. Likewise, the R05 compatible library should be merged if object compatibility is desired.

The F7RTLLIB.CSS is supplied with the Package to be used to merge the appropriate files.

F7RTLLIB.CSS accepts 4 arguments:

1. Specifies the argument checking option for the Run-Time Library (07-203), and the System Math Library (07-305F02).

A -- argument checking.

The default is no argument checking.

2. Specifies the WCS option for the System Math Library (07-305F04).

X -- 3212/3230/3240/3250/3200MPS/3260MPS WCS (07-305 F03-F04).

W -- 8/32 WCS (07-305 F05-F06).

The default is no WCS (07-305 F01-F02).

3. Specified whether to include the compatible library (07-305F04)

C -- R05 compatibility

The default is no compatibility.

4. Specifies the filename (.LIB already assumed) on which this generated RTL is to be created. It defaults to F7RTL60.LIB. Note that the Ease-of-Use (EOUF7.CSS) and F7HLOC-generated CSSs use F7RTL60.LIB in their LINK sequences.

Examples to build the RTL:

- **F7RTLLIB**

- * creates a FORTRAN VII RTL on F7RTL60.LIB without WCS and, without argument checking, and without R05 compatibility.

- **F7RTLLIB A,X,C,F7RTLXAC**

- * creates a complete argument-checking, WCS-RTL on the file F7RTLXAC.LIB which is object compatible with R05.

Please note that not all object libraries are supplied with every package. If an object library is not available to satisfy the requested F7RTLLIB command, the attempt to build the library will be terminated and a message will be issued. For example, if an attempt is made to build a 3200 Series WCS RTL and F7M60X.OBJ is not present, F7RTLLIB will terminate after issuing a warning that F7M60X.OBJ is not present.

The following table shows the libraries that can be built with the various packages:

Run Time Library	PACKAGES 04-XXX					
	149	175	176	177	180	459
NON-WCS RTL	yes					yes
SERIES 3200 WCS RTL	yes					
8/32 WCS RTL	yes					

Examples:

1. Only package 04-149 can build all variations.
2. With the package 04-459, one can only build a NON-WCS Run Time Library (Argument checking and Non-Argument checking). A WCS-RTL cannot be built with this package.

7.3 THE FORTRAN VII EASE-OF-USE PROCEDURES

This section is applicable only to the following packages:

04-149 -- Universal Fortran Language System

04-459 -- Fortran VII Compiler

The FORTRAN VII Ease-of-Use System is supplied as an update for users of the current Ease-of-Use System (03-342) which is supplied with the OS/32 Multi-Terminal Monitor (MTM) Package (04-083). This update will permit users to make the program development commands of EOU available for FORTRAN compilers. This system is not compatible with any revision of OS/32 prior to OS/32 R06-01. The FORTRAN VII High Level Operator Command (F7HLOC) procedures are supplied in the package for users of OS/32 R05-02.

Note that two versions of the Ease-of-Use CSS procedure are supplied. EOUF7.CSS is for users of OS/32 R06-02 and above, and is incompatible with OS/32 R06-01. EOU61F7.CSS and NEOUF7.TSK are supplied for users of OS/32 R06-01.

The parts supplied in this package consist of an Ease-of-Use Program Environment Monitor Task and an Ease-of-Use CSS Procedure. When the Ease-of-Use CSS is run, it will update the program development procedures generated by MTM EOU.CSS to reflect FORTRAN VII R06-00 changes.

Users with OS/32 R07-01 and above

In order to properly update the Ease-of-Use System, the user must do the following:

1. EOU.CSS, which is supplied with the OS/32 MTM package, must be run from an MTM terminal. Once started, this CSS will interactively explain its own use and assist in tailoring itself to the user's installation.
2. EOUF7.CSS must then be run to update the CSSs created above.

Users with OS/32 R06-02

In order to properly update the Ease-of-Use System, the user must do the following:

1. Delete NEOUF7.TSK and EOU61F7.CSS as these parts are incompatible with the OS/32 R06-02 EOU System.
2. EOU.CSS (03-342 F04 R02), which is supplied with the OS/32 MTM (04-083), must be run from an MTM terminal. Once started, this CSS will interactively explain its own use and assist in tailoring itself to the user's installation.
3. Rename the NEOU62.TSK as EOU.TSK.
4. EOUF7.CSS must then be run to update the CSSs created above.

Users with OS/32 R06-01

In order to properly update the Ease-of-Use System, the user must do the following:

1. The Program Environment Monitor Task (NEOUF7.TSK) must be placed on the system volume and system account.
2. EOU.CSS (03-342 F04), which is supplied with OS/32 MTM [04-083 (R06-01)], must be run from the system console with the volume set to the system volume.
3. EOU61F7.CSS must then also be run to update the existing CSS.

A description of the use of the program development facilities available can be found in the **Developing Programs with FORTRAN VII - A Guide**, Publication Number 48-010, and in the **OS/32 Multi-Terminal Monitor (MTM) Reference Manual**, Publication Number 48-023.

7.4 HIGH LEVEL OPERATOR COMMAND LEVEL (HLOC)

This section is applicable only to the following packages:

- 04-149 -- Universal Fortran Language System
- 04-459 -- Fortran VII Compiler

The FORTRAN VII HLOC procedures are supplied as part of this package to support users of OS/32 R05-02 or users OS/32 R06-01 or above who do not wish to use the FORTRAN VII Ease-of-Use procedures.

Description of F7HLOC.CSS

F7HLOC.CSS is a command substitution system file which generates the following help and CSS files:

- F7C.HLP - Help file for compilation.
- F7CE.HLP - Help file for compilation and task establishment
- F7CLG.HLP - Help file for compilation, task establishment and execution.
- F7C.CSS - CSS to compile under any F7 compiler.
- F7Z.CSS - CSS to compile under F7Z.
- F7O.CSS - CSS to compile under F7O.
- F7CE.CSS* - CSS to compile under any F7 compiler and to establish the task.
- F7CLG.CSS* - CSS to compile under any F7 compiler, to establish and to execute the task.

*NOTE: The user task is established with the F7RTL60.LIB Run-Time Library.

F7Z.CSS and F7O.CSS take five parameters.

1. FORTRAN input source filename without extension; extension is assumed to be .FTN.
2. Object filename without extension; extension is assumed to be .OBJ. The default object filename is the first parameter concatenated with .OBJ.
3. Source listing device or filename. The default is PR:
4. Start options for compilation. OPTIMIZE and BATCH are default options for the compilers in the F7O and F7Z CSSs.
5. Load size for either compiler. The default is 300KB for F7O and 300KB for F7Z.

If in-line assembler code is present, the assembler output is sent to a file with the name formed by the concatenation of .CAL either with the second parameter, if it exists, or with the first parameter. Then the resulting code is assembled using CAL/32.

F7Z.CSS and F7O.CSS attempt to allocate a scratch file of 4000 contiguous sectors for table paging. If the CSS fails to allocate 4000, the CSS will attempt to allocate 2000 sectors and then 1000 sectors. If this last attempt fails, the CSS will not assign a temporary file for paging.

F7C.CSS takes six parameters. The first is Z or O for compilation under the Universal or Global mode compilers. The next five parameters are the same as those parameters passed to F7Z.CSS or F7O.CSS.

F7CE.CSS takes 7 parameters:

1. Z or O for compilation under Universal or Global mode compilers, respectively.
2. FORTRAN input source filename. (Same as the first parameter of F7Z.CSS, F7O.CSS.)
3. Task filename without extension; extension will be defined as .TSK. The default task filename is the second parameter concatenated with .TSK.
4. Source listing device or filename. (Same as parameter 3 for F7Z.CSS, F7O.CSS.)
5. Start options for compilation.
6. Load size for compilers.
7. X if LINK R00-01 or lower is used to build the task. This option should not be specified for any release above LINK R00-01.

F7CLG.CSS takes the same 7 parameters as F7CE.CSS.

Example:

Let NIMROD.FTN contain in-line assembly code and many subprograms. A source listing with cross-reference is to be sent to the PR: (start option for cross-reference is XREF). The following invocation of F7CLG.CSS compiles with optimization, establishes and executes the task, which will be called NIMROD.TSK.

```
F7CLG O,NIMROD,,,XREF,200
```

Please refer to the **Developing Programs with FORTRAN VII - A Guide**, Publication Number 48-010, for a complete explanation of how to fully exploit the special features of the FORTRAN VII Z, and O Compilers.

7.5 ESTABLISHING THE WRITABLE CONTROL STORE (WCS) SUPPORT PROGRAMS

This section is applicable only to the following packages:

04-149 -- Universal FORTRAN Language System
04-175 -- Series 3200 FORTRAN Enhancement Package

The user must establish the writable control store (WCS) support programs as **resident, executive** tasks with a **priority higher than any other task which uses WCS**. To do so, the user must run a version of LINK compatible with the operating system installed on the user's computer. The user should enter from the system console the following sequence of commands to LINK in order to establish both versions of the WCS Support Program:

OPT PRI=(11,11)	Establish task with initial priority of 11, maximum priority of 11.
OPT ET,RES[,FL,DFL]	Make it an executive, resident task. Specify the floating point options if so equipped.
OPT ABS=0	The UDL is already included in the object file(LINK will respond with a warning which can be ignored.)
OPT WORK=50	Get storage for start options
IN LDWCS.OBJ	Include the WCS Support Program version to be established.
MAP PR:	Get a map of the task.
BU LDWCS.TSK	Build the task.
END	

This establishes the task as an executive task (E-task) with the highest priority; it is sufficient to establish it as a task with priority higher than any other task which uses WCS.

Note that the task has been established using ABS=0. Setting ABS to any other value will lead to illegal instructions when the task is run. Furthermore, if the task is established as non-resident or if the task is deleted from memory, the task must be reloaded and restarted after a power failure in order to initialize the contents of WCS. **Failure to do so will leave the contents of WCS undefined and lead to unpredictable results and system crashes.**

7.6 WRITABLE CONTROL STORE (WCS) RUN-TIME LIBRARIES (RTLs)

This section is applicable only to the following packages:

- 04-149 -- Universal FORTRAN Language System**
- 04-175 -- Series 3200 FORTRAN Enhancement Package**

There are two mathematical WCS libraries, one with argument checking and one without. It is recommended that when a program is being debugged, the task be established using the argument checking library. This will enable the user to find any illegal argument types which are being passed to any library routines.

More information on the WCS Libraries may be found in the System Math Library Reference Manual, Publication Number 48-025, and the High Speed FORTRAN Package User's Guide, Publication Number 29-643.

When the libraries are to be used with the FORTRAN VII compilers, they should first be merged with the remainder of the FORTRAN VII Run-Time Library through use of the F7RTLLIB.CSS file which is supplied with the following FORTRAN VII packages:

Package Number	Package Name
04-149 R05	Universal FORTRAN Language System Package
04-459 R00	Fortran VII Compiler Package

8. MICROCODE RUN-TIME LIBRARY (RTL) ROUTINES

This section is applicable only to the following packages:

- 04-149 -- Universal Fortran Language System
- 04-175 -- Series 3200 Fortran Enhancement Package

There are three versions of the microcode RTL routines: one, MICRTL45.WCS, for the Models 3240, 3250, 3250XP, and 3200MPS/3260MPS central processing unit (CPU), the second, MICRTL30.WCS, for the Models 3212, 3230, 3230XP, 3230MPS and the third, MICRTLAP.WCS, for the 3200MPS/3260MPS Auxiliary Processing Unit (APU). When using the 3200MPS/3260MPS, load the CPU with MICRTL45.WCS and the APU with MICRTLAP.WCS. When using a 3230MPS, load the CPU and APU with MICRTL30.WCS. These routines are supplied in the form of writable control store image files. The microcode RTL routines from the WCS image file may be loaded in the WCS of a processor using the loader and powerfail monitor (MPSLPFM) supplied with the 3200 series WCS Support Software Package, with Part number 04-174. For more information on the LPFM routines, consult the packaging documents supplied with the 04-174 package.

Recommended Microcode RTL	Processor(s)
MICRTL45.WCS	3240, 3250, 3260CPU, 3250XP
MICRTL30.WCS	3212, 3230, 3230MPS (CPU & APU), 3230XP
MICRTLAP.WCS	3260APU

The only difference between MICRTL30.WCS and MICRTLAP.WCS is in the implementation of the routine RXXR (called to raise a single precision number to a single precision power.) In the latter, RXXR is implemented using Double Precision arithmetic, the same as is done in the MICRTL45.WCS. MICRTL30.WCS implements RXXR in single precision.

9. FORTRAN VII Performance Utilities

This section is applicable only to the following packages:

- 04-149 -- Universal Fortran Language System
- 04-177 -- Fortran Performance Utilities Package

The FORTRAN VII Performance measurement support routines are Execution Profile Analyzer (XPA) and Call Recorder Analyzer (CRA). These routines are included in the object library file F7PFUT.OBJ. The modules, .XPATIMR, .CRA, or .CRAXPA, should be included from this file when building the task whose run-time performance is to be monitored. For more information on the use of CRA and XPA, refer to the **Developing Programs with FORTRAN VII - A Guide**, Publication Number 48-010.

APPENDIX A - RELEASE NOTES

This appendix contains important release notes about each product, known problems and avoidance procedures.

• FORTRAN VII O & Z

This section is applicable only to the following packages:

04-149 -- Universal Fortran Language System
04-180 -- Fortran VII Universal Compiler Upgrade
04-459 -- Fortran VII Compiler

NOTES

1. The R06 compilers have been enhanced to generate more optimal code in cases where alternate returns are used. This enhancement enables the the compiler's optimization algorithms to better determine the optimal locations to place load/store instructions for globally allocated registers. **This enhancement makes code generated by the R06 compilers to be incompatible with code generated by earlier (R05-xx) compilers.** Therefore, subprograms with alternate returns in their argument list and subprograms that call subprograms with alternate returns must be compiled by the same release of compilers, i.e., R06 or R05-xx, however, it is advised that you recompile with R06 in this situation.

KNOWN PROBLEMS

1. The optimization summary may contain erroneous information about dead code being eliminated, although this is not the case. This occurs in code sequences similar to the one below:

```
15      continue
      call xxx ()
      call zzz (i)
      if (i .ne.0 ) then
          cnt = cnt+1
          if(cnt .lt. 120) then
              go to 15
          else
              write(6,*) 'a message'
          end if
      end if
```

The optimization summary will specify that the "go to 15" and "else" statements were deleted.

Avoidance - If you encounter such messages and you determine that it the optimization is erroneous, generate an assembly listing and check to make sure the code hasn't been deleted.

2. A program with a large number of CHARACTER constant arguments passed to external programs may produce "COMPILER INTERNAL TABLE OVERFLOW" message and terminate with an "End of task code 9".

• FORTRAN VII Run-Time Library

This section is applicable only to the following packages:

- 04-149 -- Universal Fortran Language System
- 04-180 -- Fortran VII Universal Compiler Upgrade
- 04-459 -- Fortran VII Compiler

NOTES

1. **Dynamic memory management** - The I/O support routines in the R06 Run-Time Library has been rewritten in C. This was done to facilitate the addition of the new features that are part of R06 and to improve on the library's maintainability.

The use of the new library with existing applications may introduce compatibility problems for applications which use the get and release storage SVCs (SVC 2, code 2 and SVC 2, code 3). It is strongly advised to modify your code if your application uses both get and release storage SVCs. You do **not** need to modify your code if only the get storage SVC is used.

The following routines are provided to use in place of the get and release storage SVCs. Note that they are callable from the assembly level only. They are not accessible from the FORTRAN level.

• **MALLOC(NEEDED)**

MALLOC is used in place of the get storage SVC. Sample instructions needed to invoke malloc:

```
SIZE      EQU    200
GETBLK    DB     0,2
          DC     1
BLKSIZ    DAC    SIZE
          .
          .
          .
          LA    14,BLKSIZ
          BAL   15,_MALLOC
          .
          .
          .
```

Register 14 points to malloc's argument list and register 15 must be used as the link register. The address of the start of the memory allocated is always returned in general purpose register 13. This address must be recorded if you plan to restore (free) or partially restore the memory later. The value of register 13 is 0 if the allocation request failed. Note that the argument list is some arbitrary location which containing the amount of space, in bytes, requested. In the above example the location which formerly defined the size for the get storage SVC is used.

If the use of registers 13-15 presents a problem in your code then it suggested that you use a few extra instructions to provide an interface.

	SIS	1,12	Get space from the stack
	STM	13,0(1)	Save the registers
	LA	14,BLKSIZ	Set up arguments
	BAL	15,malloc	Get storage
	LR	RX,13	Move address to desired
register			
	ST	RX,SAVEADDR	Record address for FREE
	LM	13,0(1)	Get registers back
	AIS	1,12	Put space back on the stack

Note that "RX" is the register that previously received the starting address from the get storage SVC.

• **FREE(ADDRESS)**

FREE is used in place of the release storage SVC. Note that FREE isn't 100% equivalent to the release storage SVC. The release storage SVC returns storage starting from the top of dynamic storage, i.e., the storage which was last allocated. FREE releases storage starting from the address it is given in the argument. This address **must** correspond to an address which was returned after a call to MALLOC. Passing an arbitrary address will produce unpredictable results. The amount of storage release by FREE is equal to the amount allocated by the corresponding MALLOC call.

The code is similar to MALLOC's:

```

LA 14,SAVEADDR Set up the argument list
BAL 15,_FREE  Invoke it

```

Register 13 will contain a zero (0) if no errors were detected, otherwise register 13 is set to one (1). If the use of registers 13 - 15 is problematic in your code use a similar code sequence as described for MALLOC.

• **PFREE(ADDRESS,AMOUNT)**

PFREE is provided if you only want to release a partial amount of the storage which was previously allocated by MALLOC. PFREE requires two arguments; the starting address of the chunk of storage and the amount of storage to release. The storage is released starting from the top of the chunk. Note that you cannot partially release more than what was previously allocated.

Sample instructions for PFREE

```

SAVEADDR DSF 1
AMOUNT   DSF 1
.
.
.
LA 14,SAVEADDR Set up the argument list
BAL 15,_PFREE  Invoke it

```

The argument list is simply two contiguous fullwords; the first being the address and

the second the amount to release.

2. **Storage requirements** - The dynamic storage required to support the R06 RTL has increased. The formula to determine the amount of workspace needed is as follows:

$$\text{work} = 8\text{kb} + \text{I/O buffer space} + (68 * \# \text{ of Logical Units}) + 114 + 268$$

task queue size + message ring size.

8kb is the workspace required for the RTL stack

68 is the amount of space needed for each logical unit and 114 is the size of the static communication area.

268 is the maximum workspace required to support start options used by the GETOPTS routine.

Task queue size and message ring size must be factored in if the task is using the task queue service trap. The workspace required for the task queue is calculated as follows:

$$4 * \# \text{ of queue entries} + 8$$

Note that the number of queue entries default is 48. You may increase this size using the INIT routine. The message ring does not require any workspace if the default size of 2 is used. Otherwise, the workspace required for the message ring is calculated as follows:

$$76 * \# \text{ of message buffers}$$

For an explanation of the amount of I/O buffer space needed please refer to Chapter 7 of **Developing Programs with FORTRAN VII - A Guide**.

3. **Run time stack size** - The size of the run time stack can now be changed via a patch using the OS/32 PATCH utility. This may become necessary if your program uses the new REENTRANT feature of the compiler. The default size of the run time stack is X'2000'. To modify the stack apply the following patch:

```
OBJ    F7LIB60.LIB,NEWRTL.LIB,LIB
GET    __U,COPY
BIAS   O:P
VER    14,0,2000
MOD    14,0,3000
SAVE   COPY,TERM
END
```

In the above example the size of the stack was increased from 8192 to 12228. Note that the amount of workspace required by your task will increase if the size of the run-time stack is increased.

4. **Shared segments** - In the R06 release it was necessary to add COMMON entries to the initialization routine (__U) to support the RTOLBIT/LTORBIT and TRANSCENDENTAL compiler start options. Therefore, __U must **not** be included in a shared segment. To build a shared segment that contains all modules except __U, use the OS/32 Library Loader as follows:

```
*AL F7RTL.TMP,IN,128/8          <--- allocate the new library
```

```
*L LIBLDR
*AS 1,F7RTL.LIB/S,SRO          <--- assign the R06 library
*AS 2,F7RTL.TMP                <--- assign the new library
*AS 3,NULL:
*AS 5,CON:
*ST
>DU 1,2 __U                    <--- dupe all modules until __U
>CO 1,3                        <--- do a copy to go past __U
>DU 1,2                        <--- dupe the remaining modules
>END
```

Follow the directions in Chapter 7, **Linking Shared Segments**, to build the shared library.

5. Reliance Notes:

- Reliance FORTRAN programs compiled and linked under version 6 of the FORTRAN Language System require more memory to run than under version 5. If your Reliance thread size is not large enough then a FORTRAN program will fail with the message 'INVALID ATTEMPT TO CALL FORTRAN ROUTINE .V' in the Reliance journal and either the message 'INSUFFICIENT MEMORY' or the message 'UNABLE TO GET ENOUGH STORAGE TO DO I/O' on the system console.
- RELHLOC, as supplied with Reliance Plus R08.01, generates FORTRAN VII compile and link CSS files that use version 5 of the FORTRAN VII compilers. If you are using version 6 of the FORTRAN VII compilers then the following files must be updated as shown below:

RFCOMPC.CSS - change all references of 53 and 05 to 60 on lines 51, 53, 61, 73, 75, 80, 107, 109, 114.

RFLINK.CSS - change all references of F7RTLA.OBJ to F7RTLA.LIB on lines 95, 97, 112, 114. Also, for Reliance FORTRAN production programs include the line "IN F7REL60.LIB/S" before line 94 (IN REL.OBJ/S). Note that any user-created filename.LNK file must also be updated.

RFLINK.HLP - change F7RTLA.OBJ to F7RTLA.LIB on line 31.

6. Reading and Writing Past End-of-File

To fulfill the Military Standard 1753 extensions, reading and writing past an end-of-file record will be permitted for unlabeled magnetic tape sequential files. However, this feature does not conform to the ANSI '77 standard. Since reading past end-of-file is not typical of conventional programming (writing past end-of-file being typical), only this aspect is addressed below.

When reading, if the input statement (READ, ACCEPT) does not contain an "END=" or an "IOSTAT=" specifier, reading past end-of-file record is not permitted. If an "END=" or "IOSTAT=" specifier is used in a READ or ACCEPT statement and an end-of-file record is encountered, the Run-Time system will remember that an end-of-file record has been read on that unit. Only a BACKSPACE or a REWIND can reset this information to allow another read up to the same end of file record. In addition, reading past end-of-file will not cause the file size to grow (i.e. the EOF marker does not advance).

If an ENDFILE record is written onto a file connected to a unit, the file pointer is positioned after the end-of-file record. This end-of-file record is counted as a logical record for purposes of random access.

If "END=" or "IOSTAT=" specifier is specified in a READ statement, and an end-of-file record is encountered, the file is left positioned after the end-of-file record. Any subsequent input/output statement will cause a read/write beyond that end-of-file record. This operation will return whatever is in the device following the end-of-file record.

KNOWN PROBLEMS

1. When using FORTRAN VII with Reliance, if the thread is not given an adequate size, two undefined error codes are generated followed by an error code 3, "UNABLE TO GET ENOUGH STORAGE TO DO I/O".

Avoidance - increase the size of the thread.

2. The following usage of a FORMAT statement will result in unintended results:

```
      WRITE (6,100) (A(I), I=1,10)
100   FORMAT(1X,5(I7))
```

If the array contains all zeros the output would appear as:

```
      0      0      0      0      0
      0      0      0      0      0
```

The problem is that the formatter does not restart from the first parenthesis after the first record is written. Thus, in the above example, the '1X' is not included in the second record.

Avoidance - rewrite the FORMAT statement as: `FORMAT((1X,5(I7)))`

3. Backspacing an unformatted magnetic tape file whose BLOCKSIZE is less than (RECL + 4) does not backspace the expected number of records.

Avoidance - When using an OPEN statement to assign the magnetic tape, set the RECL to a minimum of (RECL + 4). For example:

```
OPEN(LU, FILE='MAG1:', RECL=1024, BLOCKSIZE=1028)
```

• FORTRAN VII Documentation

KNOWN PROBLEMS

1. Document - Developing Programs with FORTRAN VII - A Guide:

- In the section titled "Linking Shared Segments", on page 7-22, all places where the RTL routine .U is referred should be changed to __U. Also, this section should contain information stating that __U cannot be included in a shared segment. (Please refer to release note 4 of the FORTRAN VII Run-Time Library).
- In Appendix A the section titled "RTL Constants" should be replaced with the following text:

The RTL initialization routine, __U, contains an entry point (.STACKSZ) defining the size of the run-time stack set to a default size of X'2000'. You may modify the stack size with the following patch:

```
OBJ  F7LIB60.LIB,NEWRTL.LIB,LIB
GET  __U,COPY
BIAS 0:P
VER  14,0,2000
MOD  14,0,3000
SAVE COPY,TERM
END
```

In the above example the size of the stack was increased from 8192 to 12228. Note that the amount of workspace required by your task will increase if the size of the run-time stack is increased.

Note that the .CONLEN entry is no longer used. Console length is determined by the sysgen's default.

APPENDIX C - PROBLEM REPORT PROCEDURES

To report any problems, obtain Software Change Request (SCR) forms from your local Concurrent Computer Corporation office and send the SCRs to the SCR COORDINATOR of CONCURRENT COMPUTER CORPORATION, 2 Crescent Place, Oceanport, NJ 07757. Please send the SCR with:

- listing of FORTRAN source (assembly if using FORTRAN VII O or Z) and cross-reference listing
- the FORTRAN source on tape in BACKUP form
- description of the problem
- run-time data if applicable
- revision, update and patch number of the compiler used
- a listing of the LINK map of the failing program

Customers with appropriate service contracts may call 1-800-73CARES (USA only) for immediate assistance.