1 Overview

Audience

This manual is written for those who are using iWarp™ FORTRAN to develop programs on iWarp systems. The manual assumes that the reader understands the UNIX operating system and the FORTRAN programming language.

Using this manual

This manual is both a user's guide and a reference manual, and is intended for both the new and experienced iWarp FORTRAN programmer.

The manual is designed as follows:

Chapter 1:	Provides an overview of iWarp FORTRAN
Chapter 2:	Describes how to invoke the iWarp FORTRAN compiler
Chapter 3:	Explains the iWarp FORTRAN extensions
Chapter 4:	Explains how to increase the performance of your iWarp FORTRAN programs
Chapter 5:	Explains how to interface iWarp FORTRAN programs to programs in other languages
Appendix A:	Provides a quick reference to the iWarp FORTRAN extensions
Appendix B:	Describes the supported VMS extensions
Appendix C:	Lists the iWarp include files
Appendix D:	Describes the implementation specifics of iWarp FORTRAN
Appendix E:	Lists the iWarp FORTRAN error messages

Conventions:

Documentation conventions

- *italic type* indicates items you must provide.
- `courier font` is used for program and command examples.

In a syntax description:

- **[ ]** square brackets enclosing qualifiers indicate the qualifiers are optional.
- **{ }** braces enclosing qualifiers indicate that one or more of the qualifiers is required.

- | a vertical rule separating qualifiers indicates that only one of the qualifiers can be selected.
- ... a horizontal ellipsis indicates that the preceding optional item can appear one or more times in succession.
- ⋮ a vertical ellipsis indicates the continuation of statements within a program example.
- spaces are used between entities to improve readability in both syntax descriptions and programs but unless otherwise noted have no significance.

List of Related Documents

The following documents provide more information about the iWarp system.

- *Introduction to iWarp*
- *iWarp C User's Guide*
- *iWarp Programmer's Guide*
- *iWarp Debugger Manual*
- *Apply Manual*

iWarp FORTRAN Overview

The iWarp FORTRAN compiler is a port of the AT&T Portable FORTRAN 77 Compiler (PCC2 technology), with a few iWarp-specific extensions and support for VMS extensions added. The iWarp FORTRAN compiler compiles standard ANSI 77 FORTRAN source code. This manual does not describe ANSI FORTRAN. For a description of ANSI FORTRAN, refer to the *American National Standard Programming Language FORTRAN,* ANSI C3.9-1978, American National Standards Institute, New York, 1978.

Path and MANPATH Settings

To access the iWarp software and iWarp man pages, you must add /iwarp/bin to the beginning of your path and /iwarp/man to your MANPATH.
Use the following commands to do the setup (add to ~/.cshrc):

```
setenv MANPATH /iwarp/man:/usr/share/man
set path = ( /iwarp/bin $path )
```

You can now use the man command to access iWarp manpages. To invoke the man page describing iWarp FORTRAN, use the following command:

```
man iwf77
```

EFL and RATFOR Preprocessors

Intel does not distribute RATFOR or EFL preprocessors, but the iWarp FORTRAN driver does support their use. If you wish to use RATFOR (.r) and/or EFL (.e) sources with iWarp FORTRAN, your site administrator will need to make sure that you have the appropriate preprocessor installed at your site.

The iWarp FORTRAN driver assumes that the RATFOR preprocessor is named ‘‘ratfor’’ and is in the /usr/bin/ directory. A typical invocation would be the following:

```
/usr/bin/ratfor foo.r > foo.f
```

You can change the path used for ‘‘ratfor’’ with the –Yr, *path* option. This option is described in Chapter 2. For example, –Yr,/local/bin would cause the invocation for the RATFOR preprocessor to become the following:

```
/local/bin/ratfor foo.r > foo.f
```

The iWarp FORTRAN driver assumes that the EFL preprocessor is named ‘‘efl’’ and is in the /usr/bin/ directory. A typical invocation would be the following:

```
/usr/bin/efl system=unix foo.e > foo.f
```

The path used for ‘‘efl’’ can be changed using the –Ye, *path* option. Using –Ye,/local/bin would cause the invocation for the EFL preprocessor to become the following:

```
/local/bin/efl system=unix foo.e > foo.f
```