

Concurrent F77

Version 7.2 Release Notes (Linux Cross)

April 2006

0898536-7.2



Copyright

Copyright 2006 by Concurrent Computer Corporation. All rights reserved. This publication or any part thereof is intended for use with Concurrent Computer Corporation products by Concurrent Computer Corporation personnel, customers, and end-users. It may not be reproduced in any form without the written permission of the publisher.

Disclaimer

The information contained in this document is subject to change without notice. Concurrent Computer Corporation has taken efforts to remove errors from this document, however, Concurrent Computer Corporation's only liability regarding errors that may still exist is to correct said errors upon their being made known to Concurrent Computer Corporation.

License

Duplication of this manual without the written consent of Concurrent Computer Corporation is prohibited. Any copy of this manual reproduced with permission must include the Concurrent Computer Corporation copyright notice.

Trademark Acknowledgments

RedHawk, MAXAda, NightBench, PowerWorks, PowerMAXION, PowerMAX OS, TurboHawk, and Power Hawk are trademarks of Concurrent Computer Corporation.

Night Hawk is a registered trademark of Concurrent Computer Corporation.

Motorola is a registered trademark of Motorola, Inc.

PowerStack is a trademark of Motorola, Inc.

Linux is a registered trademark of Linus Torvalds.

Red Hat is a registered trademark of Red Hat, Inc.

Intel is a registered trademark of Intel Corporation.

X Window System is a trademark of The Open Group.

Contents

1.0 Introduction	1
2.0 Documentation	2
3.0 Prerequisites	3
3.1 Host System	3
3.1.1 Software	3
3.1.2 Hardware	3
3.2 Target System	3
3.2.1 Software	3
3.2.2 Hardware	3
4.0 System Installation	4
4.1 Separate Host Installation	4
4.2 Cross-Development Libraries	5
5.0 Using Concurrent F77	7
5.1 Invoking the Compiler	7
5.2 PATH and PDE_RELEASE Considerations	7
5.3 Makefile Considerations	8
5.3.1 Explicit Modification Using hf77/cf77/xf77	8
5.3.2 Use of F77 Environment or Make Variables	8
6.0 Changes in This Release	9
6.1 Control-M (Carriage Return)	9
6.2 Support for Targeting AMD64	9
6.3 INTEGER*8 and LOGICAL*8	9
6.4 -Qall_variables_in_memory	9
6.5 Driver Restructure	10
6.6 Passing COMPLEX by Value	10
6.7 -Qcommon_underscore_count=N	10
6.8 END PROGRAM, END SUBROUTINE, END FUNCTION and END BLOCKDATA	10
6.9 -W and --pass_to_... Options	10
6.10 -save-temps	10
6.11 Additional Filename Suffixes and Related Options	11
6.12 Use of '=' in --option=argument	11
6.13 -Qpower_alignment	11
6.14 Code Quality Improvements	11
7.0 Cautions	12
7.1 Byte Endian Issues	12
7.2 CPP	12
7.3 NaN	12
7.4 CALL Used on Floating-Point Functions	12
7.5 Cross Compiling	12

7.6 ASSIGN Statement	12
7.7 POINTER Blocks	13
7.8 Linking with gcc, g++ or g77	13
7.9 DWARF 3	13
8.0 Direct Software Support	14

1.0. Introduction

Concurrent F77 is based on Concurrent's Common Code Generator (CCG) technology. Concurrent F77 provides numerous switches and enhancements to provide compatibility with a number of legacy F77 compilers.

2.0. Documentation

Table 2-1 lists the Concurrent F77 7.2 documentation available from Concurrent.

Table 2-1. Concurrent F77 Version 7.2 Documentation

Manual Name	Pub. Number
<i>hf77 Fortran Reference Manual</i>	0890240-100
<i>Concurrent F77 Version 7.2 Release Notes (Linux Cross)</i>	0898536-7.2

Copies of the Concurrent documentation can be ordered by contacting the Concurrent Software Support Center. The toll-free number for calls within the continental United States is 1-800-245-6453. For calls outside the continental United States, the number is 1-954-283-1822.

Additionally, the manuals listed above are available:

- online using the PowerWorks Linux Development Environment utility, **nhelp** (*release notes only*)
- in PDF format in the **documentation** directory of the PLDE Installation CD
- on the Concurrent Computer Corporation web site at **www.ccur.com**

3.0. Prerequisites

Prerequisites for Concurrent F77 Version 7.2 for both the host system and target system are as follows:

3.1. Host System

3.1.1. Software

- Any of the following operating systems:
 - Red Hat® Linux® 8.0 or 9.0
 - Red Hat Enterprise Linux WS 3.0 or 4.0
 - RedHawk™ Linux 1.4 or later
- X Window System¹
- Cross development package(s) for the target PowerMAX OS™ system (see “Cross-Development Libraries” on page 5)

3.1.2. Hardware

- an Intel®-based PC - 300Mhz or higher (recommended minimum configuration)
- 64MB physical memory (recommended minimum configuration)

3.2. Target System

3.2.1. Software

- PowerMAX OS 4.3 or later

3.2.2. Hardware

- Computer Systems:
 - Power Hawk™ 620 and 640
 - Power Hawk 710, 720 and 740
 - Power Hawk 910, 920 and 940
 - PowerStack™ II and III
 - Night Hawk® Series 6000
 - TurboHawk™
 - PowerMAXION™
- Board-Level Products:
 - Motorola® MVME2604
 - Motorola MVME4604

¹ This is a prerequisite only for the the online help package. It is not necessary for proper operation of the compiler. See “System Installation” on page 4 for more information about these packages.

4.0. System Installation

Installation of the host portion of Concurrent F77 is normally done as part of the general installation of the PowerWorks Linux Development Environment software suite. A single command installs (or uninstalls) all software components of the PLDE, as described in the *PowerWorks Linux Development Environment Release Notes* (0898000).

The following section describes how to install (or uninstall) Concurrent F77 separately from the PLDE suite for those rare cases when this is necessary.

4.1. Separate Host Installation

In rare cases, it may be necessary to install (or uninstall) Concurrent F77 independent of the installation of the PowerWorks Linux Development Environment software suite. This may be done using the standard Linux product installation mechanism, `rpm` (see `rpm(8)`).

The name of the RPMs associated with Concurrent F77 7.2 are:

```
ccur-invoker
ccur-f77-manpages-7.2
ccur-f77-tools-7.2
ccur-f77help-7.2
plde-f77-7.2
plde-f77help-7.2
plde-f77libs-7.2
```

and the files associated with these RPMs are:

```
ccur-invoker-000-*.i386.rpm
ccur-f77-manpages-7.2-000-*.i386.rpm
ccur-f77-tools-7.2-000-*.i386.rpm
ccur-f77help-7.2-000-*.i386.rpm
plde-f77-7.2-000-*.i386.rpm
plde-f77help-7.2-000-*.i386.rpm
plde-f77libs-7.2-000-*.i386.rpm
```

which can be found on the PowerWorks Linux Development Environment Installation CD.

NOTE

The package `plde-f77help-7.2` contains the online manuals for the F77 compiler. The installation of `plde-f77help-7.2` is not necessary for the proper operation of the compiler.

NOTE

The user must be root in order to use the `rpm` product installation mechanism on the Linux system.

To install the Concurrent F77 RPMs, issue the following commands on your Linux system in the following order:

```
rpm -i ccur-invoker-000-*.i386.rpm
rpm -i plde-f77libs-7.2-000-*.i386.rpm
rpm -i ccur-f77-manpages-7.2-000-*.i386.rpm
rpm -i ccur-f77-tools-7.2-000-*.i386.rpm
rpm -i plde-f77-7.2-000-*.i386.rpm
rpm -i ccur-f77help-7.2-000-*.i386.rpm
rpm -i plde-f77help-7.2-000-*.i386.rpm
```

NOTE

The **plde-f77help-7.2** RPM is not required for the compiler to function.

To uninstall the Concurrent F77 RPM, use the following commands:

```
rpm -e plde-f77help-7.2
rpm -e ccur-f77help-7.2
rpm -e plde-f77-7.2
rpm -e ccur-f77-tools-7.2
rpm -e ccur-f77-manpages-7.2
rpm -e plde-f77libs-7.2
rpm -e ccur-invoker
```

NOTE

Some RPMs are shared by multiple products. The cross C++ compiler and the native Linux Concurrent F77 compiler both use **ccur-invoker**. The native Linux Concurrent F77 compiler also uses **ccur-f77-manpages-7.2**, **ccur-f77-tools-7.2**, and **ccur-f77help-7.2**. You may need to omit the uninstall commands for these RPMs if either of these compilers are installed on the system.

4.2. Cross-Development Libraries

In order to cross-compile and cross-link on the Linux system, it is necessary to have certain PowerMAX OS libraries installed on that system. These libraries are normally installed as part of the PowerWorks Linux Development Environment.

The names of the RPMs containing the PowerMAX OS libraries minimally required for cross-linking are:

```
plde-pmax-crossdev-version
```

NOTE

The version number is part of the name of the RPM. Because of that, it is possible to install multiple RPMs on the Linux system at the same time. This allows the user to generate executables for multiple PowerMAX OS versions from the same Linux system.

User applications built with Concurrent F77 may require other capabilities which are provided by additional RPMs included on the PLDE Installation CD. Refer to the section titled “Cross-Development Libraries and Headers” in the *PowerWorks Linux Development Environment Release Notes* (0898000) for more information.

5.0. Using Concurrent F77

The following should be taken into consideration when using Concurrent F77.

5.1. Invoking the Compiler

There are three Concurrent Fortran 77 products available separately. Each is invoked differently.

- Concurrent Fortran 77 for PowerMAX (PowerMAX)
Native PowerPC™ compiler
Invoked with the **f77** or **hf77** commands (as has always been the case).
- PLDE Concurrent Fortran 77 for PowerMAX (Linux)
Intel®-to-PowerPC cross-compiler
Invoked with the **xf77** command.
- Concurrent F77 for RedHawk Linux (Linux)
Native Intel Pentium® and AMD64 compiler
Invoked with the **cf77** command (due to the fact that **f77** and **hf77** already exist on Linux systems).

5.2. PATH and PDE_RELEASE Considerations

To utilize the Concurrent F77 compiler, specify the following in your PATH environment variable:

```
PATH=$PATH:/usr/ccs/bin
```

The compiler should then be invoked with **cf77**. (This contrasts with PLDE cross compiler product which is invoked with **xf77**, or with the native PowerPC product which is invoked with **hf77** or **f77**).

Installing 7.2 compilers on your system will make 7.2 the default release. If the default PLDE release on your system is not 7.2, then it will be necessary to select the 7.2 release by one of the following methods:

- Use the **--rel=7.2** option on the **hf77/cf77/xf77** command line
- Set the **PDE_RELEASE** environment variable to **7.2**
- Set the user default release (**c.release -rel 7.2 -u**)
- Set the system default release (**c.install -rel 7.2 -d**)

5.3. Makefile Considerations

Makefiles may already contain references to **f77**, **hf77** or **g77** commands explicitly within them. Additionally, if default rules for compilation, such as

```
.f.o:
```

are not explicitly mentioned, the **make** processor will also attempt to invoke **hf77**, **f77** or **g77**.

To invoke the Concurrent F77 compilers you can take any of the following approaches.

5.3.1. Explicit Modification Using hf77/cf77/xf77

Ensure that **/usr/ccs/bin** is in your **PATH** environment variable.

Modify all occurrences of **f77**, **hf77**, and **g77** to utilize **hf77**, **cf77**, or **xf77** as appropriate.

Supply default **.f.o** rules (and the like) to explicitly utilize the appropriate command.

5.3.2. Use of F77 Environment or Make Variables

One approach that requires minimal changes to Makefiles, etc., is to use environment variables or **make** variables to control which compiler you're using. The following commands will all build using the Concurrent Native Intel F77:

Short-lived environment variables:

```
# F77=cf77 make arguments
```

make variables:

```
# make arguments F77=cf77
```

Long-lived environment variables:

```
# export F77=cf77
# make arguments
```

You can also use the long-lived environment variable approach by adding the following to your login script (e.g. **.profile** or **.login** depending on your shell):

```
export F77=cf77
```

Or, if you prefer finer-grained control, you can add lines like the following to the top of any Makefiles that should use the Concurrent F77 compiler:

```
F77=cf77
```

The changes will then only affect the modified Makefiles. Note that this solution only works for Makefiles that use the default **.f.o** rules. If they contain hard-coded references to **f77** or **g77**, then the Makefiles must be changed to use $\$(F77)$ instead. If the Makefile references anything like **g77** (Linux's GNU F77 compiler), then it will need to be changed, regardless.

6.0. Changes in This Release

The following changes were made in Concurrent F77 Version 7.2.

6.1. Control-M (Carriage Return)

Source files may end in Carriage Return-Line Feed now instead of just Line Feed.

6.2. Support for Targeting AMD64

This release has support for AMD64 (a 64-bit processor supporting the x86_64 instruction architecture). The installation CD (available separately from the PowerPC targeting compilers) contains both the i386 (32-bit) and the x86_64 (64-bit) compilers. The installation script selects the appropriate one to install on the host machine. Cross targeting the 32-bit and 64-bit instruction architectures from one another is not supported, so the user should not manually install both compilers on a single machine.

AMD64 specific libraries are in `/usr/ccs/release/7.1/lib_amd64`. Pentium specific libraries are in `/usr/ccs/release/7.1/lib_ia32`.

6.3. INTEGER*8 and LOGICAL*8

Since the x86_64 instruction architecture requires 8 bytes for storing pointers, the compiler has been enhanced to support 8-byte versions of the scalar types. All the compilers (i386, x86_64, and PowerPC) now support these new types. The default size for INTEGER and LOGICAL remains four bytes in all compilers.

The existing intrinsics and I/O statements have been extended to support the new times, and two new intrinsics, %INT8 and %LOG8 have been added. See the man pages for details.

See “ASSIGN Statement” on page 12 and “POINTER Blocks” on page 13 for cautions regarding the ASSIGN statement and POINTER blocks.

6.4. -Qall_variables_in_memory

This customer-requested option forces all variables to reside in memory.

WARNING

Since this suppresses holding variables in hardware registers, code quality will be significantly impacted.

This option may be helpful in debugging problems in user code, as it allows setting address traps on all variables.

6.5. Driver Restructure

The driver programs have been restructured so that all options are parsed before beginning the compilation. For example, you can now supply the `-c` option as the last option on the command line.

6.6. Passing COMPLEX by Value

Complex values passed with the `%VAL` intrinsic are now passed as though they were a C structure, consistent with the manual and with the Intel ABI. Previously, they were passed as two float variables. This may require some source changes.

6.7. `-Qcommon_underscore_count=N`

This customer-requested option allows the user to set the number of underscores appended to `COMMON` block names to 0, 1, or 2. The default is 2.

6.8. `END PROGRAM`, `END SUBROUTINE`, `END FUNCTION` and `END BLOCKDATA`

These new customer-requested keywords are aliases for the `END` statement closing a `PROGRAM`, `SUBROUTINE`, `FUNCTION`, or `BLOCKDATA` statement, respectively.

6.9. `-W` and `--pass_to_...` Options

Options were added to pass options to various sub-processors. Their syntax is:

```
-w{C|I|r|a|c|l|x|m|R|z}{,| }arglist  
--pass_to_cpp{=| }arglist  
--pass_to_f77{=| }arglist  
--pass_to_reorder{=| }arglist  
--pass_to_as{=| }arglist  
--pass_to_cc{=| }arglist  
--pass_to_linker{=| }arglist  
--pass_to_xref{=| }arglist  
--pass_to_m4{=| }arglist  
--pass_to_ratfor{=| }arglist  
--pass_to_analyze{=| }arglist
```

6.10. `-save-temps`

This option directs the compiler to not delete its temporary files.

6.11. Additional Filename Suffixes and Related Options

The **.FOR** filename suffix is an alias for the **.f** filename suffix.

The **.FPP** and **.fpp** filename suffixes are aliases for the **.F** filename suffix.

The **.S** filename suffix will preprocess assembly language files before passing them to the assembler.

The **.r.s** filename suffix will pass files to the **reorder** tool before being passed them to the assembler.

The **-x** {**c|assembler|assembler-with-cpp|f77|f77-cpp-input|ratfor|xref|reorder|none**} will override the the filename suffixes.

The **--stop_before_reorder** option will halt the compilation processes before invoking the **reorder** tool.

6.12. Use of '=' in *--option=argument*

Either '=' or ' ' (space) may be used to separate double-dash options from their argument.

6.13. -Qpower_alignment

This option causes the compiler to set the alignments of bss and common segments to be the same as was generated for the PowerPC. Use of this option requires all source to be recompiled.

6.14. Code Quality Improvements

A number of code quality improvements have been made, especially in the areas of COMPLEX arithmetic and constant folding transcendental functions.

7.0. Cautions

7.1. Byte Endian Issues

FORTTRAN source code that makes assumptions about byte order may require modification to run on both PowerPC and Intel processors. Such dependencies usually occur in EQUIVALENCE statements, Hollerith constants, intrinsics that extract bits, and passing by reference wrong-sized integer variables to functions and subroutines.

For example, consider code that passes an `INTEGER*4` to a subroutine expecting an `INTEGER*1`. Since the parameter is passed by reference, on PowerPC the subroutine gets a pointer to the high-order byte of the parameter, but on Intel processors, the subroutine gets a pointer to the low-order byte of the parameter.

7.2. CPP

The FORTRAN compiler uses the system `cpp` tool. There may be subtle incompatibilities between `/lib/cpp` on PowerMAX OS and `/usr/bin/cpp -traditional` on Linux.

7.3. NaN

Programs that generate a Not a Number (NaN) may cause a signal when compiled for Intel processors where they do not on the PowerPC. This is because PowerPC allows a NaN to be loaded into a floating point register without generating a signal.

7.4. CALL Used on Floating-Point Functions

On Pentium, if the `CALL` statement is used to call a function that returns a floating-point value, the F77 compiler has no way of knowing that that function has pushed a value onto the floating-point register stack. This will usually result in a floating-point register stack overflow signal being raised.

On AMD64, the calling convention does not use the floating-point register stack, so this will not be a problem for that target.

7.5. Cross Compiling

Cross compiling from `x86_64` to `i386` or from `i386` to `x86_64` is not supported. The installation script will select the appropriate compiler. Do not manually install the inappropriate compiler.

7.6. ASSIGN Statement

On AMD64 (`x86_64`), the `ASSIGN` statement must still use the 4-byte `INTEGER` type for its target variable as this is required by the standard. In actuality, an unnamed 8-byte `INTEGER*8` shadows each `INTEGER` variable used as the target of an `ASSIGN` statement and by `GOTO` and I/O statements that use

ASSIGNED to INTEGERS. The standard allows this by prohibiting intermixing ASSIGNED uses of an INTEGER variable with arithmetic uses of the same variable. There is no way at this time to access the INTEGER*8 shadow variable in the NightView debugger.

7.7. POINTER Blocks

On AMD64 (x86_64), POINTER blocks must be based on INTEGER*8 variables since pointers are 8-bytes. This is an unavoidable source incompatibility between the i386 and x86_64 compilers.

7.8. Linking with gcc, g++ or g77

This release ships with its own updated linker (**ld**) that can interpret DWARF 3. If you link **cf77** generated code using the **gcc**, **g++**, or **g77** commands, they should be directed to use this alternative **ld**. This may be done by setting the `COMPILER_PATH` environment variable to one of the following values:

```
COMPILER_PATH=/usr/ccs/release/7.2/lib_ia32
COMPILER_PATH=/usr/ccs/release/7.2/lib_amd64
```

for 32-bit Pentium code or 64-bit AMD64 code, respectively. It important that the correct path be chosen.

To make the `COMPILER_PATH` effective for the duration of a shell invocation, it can be exported:

```
export COMPILER_PATH=...
```

Or the variable may be set for a single invocation:

```
COMPILER_PATH=... gcc ...
```

If the `COMPILER_PATH` environment variable is omitted, the default system linker will be used; it will work on correct programs, but error diagnostics may be inaccurate.

The **cf77** command will use the alternative **ld** without requiring the user to set the `COMPILER_PATH` environment variable.

7.9. DWARF 3

The debug information makes use of DWARF 3. As of this writing, **gdb** does not support this update to DWARF. NightView is required to debug code from this release.

8.0. Direct Software Support

Software support is available from a central source. If you need assistance or information about your system, please contact the Concurrent Software Support Center at our toll free number 1-800-245-6453. For calls outside the continental United States, the number is 1-954-283-1822. The Software Support Center operates Monday through Friday from 8 a.m. to 5 p.m., Eastern Standard Time.

You may also submit a request for assistance at any time by using the Concurrent Computer Corporation web site at http://www.ccur.com/isd_support_contact.asp or by sending an email to support@ccur.com.

