

**Release Notice**  
**CONVEX ALL V1.1**  
Document No. 720-004930-001

---

June 1992

**CONVEX Computer Corporation**  
Richardson, Texas USA

*Release Notice, CONVEX ALL V1.1*

© 1992 CONVEX Computer Corporation  
All rights reserved.

This document is copyrighted. This document, however, may be copied, duplicated, reproduced, translated, stored electronically, or reduced to machine-readable form without prior written consent from CONVEX Computer Corporation.

Although the material contained herein has been carefully reviewed, CONVEX Computer Corporation (CONVEX) does not warrant it to be free of errors or omissions. CONVEX reserves the right to make corrections, updates, revisions or changes to the information contained herein. CONVEX does not warrant the material described herein to be free of patent infringement.

UNLESS PROVIDED OTHERWISE IN WRITING WITH CONVEX COMPUTER CORPORATION (CONVEX), THE PROGRAM DESCRIBED HEREIN IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES. THE ABOVE EXCLUSION MAY NOT BE APPLICABLE TO ALL PURCHASERS BECAUSE WARRANTY RIGHTS CAN VARY FROM STATE TO STATE. IN NO EVENT WILL CONVEX BE LIABLE TO ANYONE FOR SPECIAL, COLLATERAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING ANY LOST PROFITS OR LOST SAVINGS, ARISING OUT OF THE USE OR INABILITY TO USE THIS PROGRAM. CONVEX WILL NOT BE LIABLE EVEN IF IT HAS BEEN NOTIFIED OF THE POSSIBILITY OF SUCH DAMAGE BY THE PURCHASER OR ANY THIRD PARTY.

C200 Series architecture is a registered trademark of CONVEX Computer Corporation.  
CONVEX and the CONVEX logo ("C") are registered trademarks of CONVEX Computer Corporation.  
COVUE is a trademark of CONVEX Computer Corporation.  
Apollo is a trademark of Apollo Computer, Inc.  
Cray is a trademark of Cray Research, Inc.  
IBM is a trademark of International Business Machines Corporation.  
IEEE is a trademark of Institute of Electrical and Electronics Engineers, Inc.  
Sun is a trademark of Sun Microsystems, Inc.  
UNIX is a registered trademark of UNIX System Laboratories, Inc.  
VAX and VAX FORTRAN are trademarks of Digital Equipment Corporation.

Printed in the United States of America

CONVEX Computer Corporation

June 1992

**1****Introduction**

This document describes CONVEX ALL V1.1. It obsoletes all previous Release and Update Notices. This document is intended to enhance and clarify the existing permanent documentation for this product with information that is up-to-the-minute or was developed too late for inclusion in the permanent documentation. Always refer to this release notice before reporting questions or problems with CONVEX ALL utilities; your questions may be answered here.

The remaining sections in this document describe the contents of this release:

- Section 2 describes the contents of this distribution.
- Section 3 contains notes and warnings about the use of the software.
- Section 4 describes how to use this software.
- Section 5 describes changes to the functionality.
- Section 6 lists fixes for previously reported problems.
- Section 7 lists known problems.

CONVEX ALL is a set of language utilities and the FORTRAN and C runtime libraries. ALL consists of the CONVEX assembler (`as`), CONVEX loader (`ld`) and the CONVEX FORTRAN and CONVEX C run-time libraries (`libI77.a`, `libU77.a`, ...).



## 2

## Contents of this distribution

The distribution package for this release of CONVEX ALL consists of software and documentation. The contents of the software and documentation are described in the following tables:

### ALL software media

| Item | Qty | Type | Part Number    | Description     | Format |
|------|-----|------|----------------|-----------------|--------|
| 1    | 1   | Mag. | 720-003515-001 | CONVEX ALL V1.1 | GIP    |

### CONVEX ALL documentation

| Item | Qty | Type | Part Number    | Description                             |
|------|-----|------|----------------|---|
| 1    | 1   | Doc. | 720-005030-001 | CONVEX ALL Installation Procedure, V1.1 |
| 2    | 1   | Doc. | 720-004930-001 | CONVEX ALL Release Notes, V1.1          |



**3****Notes and warnings**

This section contains general information and words of caution about the product.

- This release requires the installation of CONVEX Utilities V9.1 or higher, and the ConvexOS Operating System V9.1 or higher.
- To install the FORTRAN libraries, this release requires that you have installed CONVEX FORTRAN V7.0 or higher.
- To install the C bit and math libraries, this release requires that you have installed CONVEX C V4.3 or higher.
- If you are doing an initial install of the operating system, you must install ALL before you do a sysgen or before you install any of the compilers.
- You will need approximately 26 megabytes of disk space in both /tmp and /usr/lib to install CONVEX ALL V1.1.



**4****Using ALL 1.1**

The assembler (as) and the loader (ld) are installed in /usr/bin with symbolic links to them from /bin. All of the CONVEX FORTRAN libraries and the CONVEX C bit and math libraries are installed in /usr/lib. All of the lint libraries are installed in /usr/lib/lint.



## Software changes and enhancements

### Libraries

The FORTRAN libraries, the C bit and math intrinsic libraries, and the lint libraries are now being released as part of ALL.

### Faster FORTRAN I/O

The FORTRAN I/O routines have been modified to increase performance. The performance increase over previous releases of the FORTRAN I/O routines is roughly 20 percent.

### Loader common block alignment option

The `-align` option has been added to the loader to align common blocks on page boundaries so that they can be mapped into the program by the `mmap(2)` system call.

### Product id stamp option

The `-nid` option has been added to the loader to prevent it from generating the product id stamp at the beginning of the `.text` section.

### Loader and multiply defined symbols

The `-w` option allows the loader to generate an executable even if there are multiply defined symbols. The symbol is defined by the first object file in which it is encountered.

### Loader and uninitialized memory

The `-init` option allows the loader to initialize uninitialized memory with a user specified value. This option is useful in debugging programs.



## 6

## Corrected problems

## Assembler

X-09378  
PR-07170

If a line with a syntax error is encountered, the assembler does not output it to `stdout` when `-l` is used. Rather, it starts the listing of the next line immediately to the right of the line number of the erroneous line. Other assemblers output the full text of the offending line, usually with some sort of flag in the area where the numeric representation of the assembled code would occur, indicating that an error had occurred.

## Resolution:

The assembler has been fixed to generate the correct source listing with the `-l` option when an error has occurred.

X-14717  
PR-14355

The assembler accepts the following assembly language code, but it should generate an error message for the label `idata`. This is because you cannot have a local label in a global region.

```

        .comm    __block_,8
        .cdata  __block_
idata:  bs.w    1

        .text
        .globl  __sub_

__sub_:
        ld.w    #-321, s0
        st.w    s0, idata
        rtn

```

## Resolution:

The assembler will now reject the above code.

X-21314  
PR-22085

The `as` coredumps upon read error. It should check for read or write failures and call `perror` if there is an error.

## Resolution:

The assembler has been updated to check for all types of I/O errors. It now does the following:

- Prints its own error message
- Calls `perror` to print the error message associated with `errno`
- Calls `abort` and exits with an IO error (core dump)

X-22294  
PR-23327

Utility `as` dies when 2 processes simultaneously try to assemble the same `.s` file. The message returned from `as` is:

```
Assembler Error--cannot reopen relocation file ../code/iwall.s_25149.tmpr.
```

The temporary file is always numbered `.s_25149.tmpr`, the `pid` is not being used to build this file name.

**Resolution:**

The assembler has now been changed to use `pid` instead of using `uid` for generating temporary filenames.

X-23640  
PR-24816

The assembler needs an `ASOPTIONS` environment variable that functions just like the current `FCOPTIONS` or `CCOPTIONS` environment variables.

**Resolution:**

This request has been granted. The assembler will now use options set in the `ASOPTIONS` environment before processing the command line options.

X-23692  
PR-24883

The assembler does not accept `LL`, `L`, `ll`, or `l` suffixes as documented in the Assembler User's Guide.

**Resolution:**

The assembler has been fixed to now accept the `LL`, `L`, `ll`, or `l` suffixes on integral constants.

X-23779  
PR-24981

The use of the `-tm c1` and `-cvt dc3` options cause the same error message to be printed multiple times for the same line of code.

**Resolution:**

The error messages associated with 'invalid instruction for target' have been changed to be consistent with other assembler error messages (the instruction that caused the error is no longer printed).

X-23829  
PR-25039

The assembler says "Reopen failed" when it cannot open a source file. This message may confuse users.

**Resolution:**

The error message has been changed to be less confusing.

X-24912  
PR-26325

The use of the `.pad` directive can cause incorrect 'multiply defined symbol' error messages to be generated with the field-test assembler.

Resolution:  
This problem has been fixed.

## Loader

X-11108  
PR-09410

This is an enhancement request to have `ld` order FORTRAN common blocks as it sees them.

Resolution:  
This request is being denied because by the definition of a FORTRAN common block it is independent of other common blocks. However we are adding a `-align` option to `ld` to align common blocks to a page boundary and pad them to a multiple of a page size.

X-13649  
PR-12827

A loader option to align FORTRAN COMMON blocks on page boundaries and pad them to page boundaries has been requested.

Resolution:  
This request has been granted. The `-align` option to the loader will now place the specified common block on a page boundary and pad it to a multiple of a page size so that it can be used with `mmap`.

X-18607  
PR-18854  
PR-25250

This is a request for a change in loader behavior. When two object files are specified on the command line and each object file contains the same procedure symbol, the loader aborts due to multiply defined symbols. On other platforms (Apollo, IBM, for two), a warning is issued and the load proceeds using the first object file encountered.

Resolution:  
This enhancement has been granted. A `-w` option has been added to the loader to tell it to treat multiply defined symbols as a warning and not as an error.

X-19635  
PR-17463  
PR-20036

Many users would like an option on the loader to initialize memory to a user defined value (or at least a choice of values which includes a reserved operand). This would aid in the debug process.

Resolution:  
This request has been granted. The loader will have an option that will allow you to initialize part or all of `.bss` with a user specified bit pattern.

X-20123  
PR-20599

In the `m1C.c` C code a data structure containing the 498 state names required for each instance of a function.

After compile this code using

```
cc -g -c m1C.c
```

then loading this code using

```
cc -g -o cn2 linkage.o m1C.o cvan_lib.a n2.ar
```

One gets

```
ld: ERROR - String '_storage:t52=s2416 ... ;c_sst:1,19296,32;;'  
too long to save (12561 characters)  
ld: No executable produced.
```

Resolution:

This problem has been fixed.

X-21178  
PR-21926

Trying to preload objects containing TASK COMMON blocks causes the loader to issue an internal error message similar to the following:

```
ld: INTERNAL ERROR - Invalid symbol type (__comm1_22) in reloc_one_symbol()  
ld: No executable produced.
```

Resolution:

This problem has been fixed.

X-21383  
PR-22154

The `ld` utility should check for read and write failures and call `perror` if there is an error.

Resolution:

All of the I/O operation do check for errors and will call `perror` in addition to the normal `ld` error routine.

X-21985  
PR-22923

The loader will allow you to specify both `-r` and `-s` options together. This should be an error because the new object file that is generated has no symbol information to be relocated later.

Resolution:

The loader will now exit with an error if you specified both `-s` and `-r` on the command line at the same time.

X-22030  
PR-22964

Currently the loader will put a product id at the beginning of the text section. There needs to be an option that prevents the loader from doing this.

Resolution:

A `-nid` option has been added to the loader to prevent it from generating the product id stamp at the beginning of the `.text` section.

X-24024  
PR-25283

An error message for `-h` option of `ld(1)` isn't reasonable because it does nothing.

Resolution:

The `-h` option has been removed from the loader. Now if you use it you will get a error message.

X-24067  
PR-25329

Sole `-p` option of `ld` causes INTERNAL ERROR.

Resolution:

This has been fixed. The loader in this case will now exit with the following error message:

```
Unresolved symbols still remain.
```

```
symbol          referenced in
```

```
foo             foo.o
```

```
ld: ERROR - incomplete load ld: No executable produced.
```

X-24841  
PR-26236

The loader overwrites `COMMON` block data initialized in a block data subprogram, if the subprogram has been compiled and placed in an object library.

Resolution:

This problem has been fixed.

## C libraries

X-22416  
PR-15162  
PR-23447

In `-ext` mode, `ipow` and `lpow` should act like all the other math routines and set `errno` to either `EDOMAIN` or `ERANGE` and not print an error message. Currently it prints an error message and sets `errno` to 304.

**Resolution:**

Both `ipow` and `lpow` now set `errno` to `EDOM` on a domain error.

X-24892  
PR-26310

The ANSI C version of the `ircvtr` function returns wrong values when you use the non-inline version of this function (`-D__NO_INLINE_MATH`).

**Resolution:**

This has been fixed.

X-23639  
PR-24813

User requests the IEEE math functions `isnan`, `isinf`, etc. that are defined in the Sun Microsystems math library. These functions are important when porting code from the Sun world that exploits IEEE not-a-numbers and infinities.

**Resolution:**

Both `isnan` and `isinf` have been added to the C libraries and are available in `-pcc` and `-ext` modes of the CONVEX C compiler. Also `isnanf` and `isinf` (single precision versions) of these functions have been added. The single precision versions are only available in `-ext` mode of the CONVEX C compiler. These new functions are documented in the `ieee_functions(3m)` man page.

X-24891  
PR-26309

The ANSI C version of the `rcvtir` function returns wrong values when you use the non-inline version of this function (`-D__NO_INLINE_MATH`).

**Resolution:**

This problem has been fixed.

## FORTRAN libraries

X-09119  
PR-06898

The `stat(3f)` routines (`stat`, `lstat` and `fstat`) only return 12 items.

**Resolution:**

The `stat(3f)` routines have been updated to return `st_blocks` as the thirteenth element of the `stat` buffer.

X-09422  
PR-07298

The `eotf` value returned from `TSTATE` does not detect physical end-of-tape. It only detects logical EOT. This makes tape error condition handling difficult. In order to check for physical end-of-tape, the `dsreg` value must be used. The man page for `mtio` describes the values for `dsreg`. It would be very nice if the `eotf` value reflected both logical and physical end-of-tape conditions.

**Resolution:**

When you call `tstate` after reaching the physical EOT, `eotf` will be set to `TRUE`.

X-09443  
PR-07029

This is an enhancement request for the `libU77/system` routine to be able to use a different shell other than the one defined in the `SHELL` environment variable.

**Resolution:**

The function `setenv` has been added to the FORTRAN libraries to allow you to change or add environment variables. Using this function you can change the default shell for system by changing the environment variable `SHELL` to whatever you want.

X-11727  
PR-10103  
PR-21332

A blank line is appended after `$END` for namelist writes. This causes problems for certain applications which do both namelist I/O and formatted I/O to or from the same file.

**Resolution:**

This request has been granted.

X-12032  
PR-10684

The FORTRAN `open` will return incorrect error information when run under `COVUEshell`. The error information will be valid under UNIX shells. Under `COVUE` status will be 121 and stack information will show the filename as `' '`.

**Resolution:**

The FORTRAN `open` routines have been fixed to return the correct status when linked or run under the `COVUE` shell.

X-13070  
PR-11932

Output from a write of `NAMELIST` starts in column 1 which causes problems if the file containing this `NAMELIST` output contains FORTRAN carriage control.

**Resolution:**

This request has been granted.

X-13482  
PR-12423  
PR-24500

The *filename.ext;0* convention of VMS to designate the most recent version of file *filename* explicitly, is not recognized by INQUIRE. INQUIRE looks for an actual filename ending in ;0. On the other hand the OPEN function works properly with this convention and opens the most recent version.

Resolution:  
This problem has been fixed.

X-13878  
PR-13164

The FORTRAN tape I/O function `tstate` does not always reflect the correct status and position of the tape.

Resolution:  
The above problem has been fixed.

X-14177  
PR-13608  
PR-13612

The compiler does not handle the non-ANSI VAX FORTRAN extension of allowing the same character positions to be referenced on both the left and right hand sides of a character assignment statement. It is requested that the compiler be modified to perform like the VAX. `name = 'abcde' // name(1:3) // 'fghij' // name(12:18)`

Resolution:  
This request has been granted.

X-14350  
PR-13748

Attempting to backspace (via `tskipr`) over a EOF marker is not properly handled when using asynchronous tape I/O.

Resolution:  
`tstate(3f)` and `tstate_lstat(3f)` have both been fixed to report correct position of the tape regardless of whether you are doing asynchronous I/O on that unit or not.

X-14578  
PR-14191

The compiler creates an executable which aborts with an integer overflow error.

```

program test
integer status, traper
character *24 fdt
status = traper('7'X)
call fdate(fdt)
write (6,*) fdt
end

```

Resolution:  
The problem is within `fdate`. The `fdate` function has been changed to not abort when integer overflow is turned on.

X-14625  
PR-14161

It would be useful to have FORTRAN functions such as the C-language functions `setbuf` and `setlinebuf` for adjusting the I/O buffer size.

**Resolution:**

This request has been granted. The function `for$flush` has been added to the FORTRAN I/O library.

X-14630  
PR-14257

User requests the ability to read past an end-of-tape (EOT) marker. If a user writes an empty file (an EOT marker) between two non-empty files, the second non-empty file cannot be read.

**Resolution:**

If you call `tignoreeot` with a second argument as true, it will allow you to read past logical or physical EOT.

X-17565  
PR-17504

When a write-only file (mode 200) is opened and a READ is attempted, an appropriate error message is issued, but the contents of the file are destroyed.

**Resolution:**

The above problem has been fixed. The open will succeed without truncating the file, and when you do a read, you will get an I/O error.

X-20112  
PR-20584

It has been requested that the relational operators handle non-printable characters that fall outside of the ASCII character set.

```

      character*1 a,b
      a=char(147)
      b=char(125)
      i=ichar(a)
      j=ichar(b)
      write (*, 100) 'i => ', i, ' j => ', j
100  format (a, i3, a, i3)
      if (a .lt. b) write (*, 200) i,j
      if (b .lt. a) write (*, 200) j,i
200  format (i3, ' is less than ', i3)
      end

```

**Resolution:**

The FORTRAN runtime character comparison routines have been modified to handle non-printable ASCII characters.

X-20329  
PR-20845

When `REWIND` is called from a FORTRAN program compiled with `-cfc` that performs asynchronous I/O using `BUFFER OUT`, it dumps core giving the following error:

```
rewind: [139] synchronous I/O not allowed on this file
```

The *CONVEX FORTRAN Language Reference Manual* does not state that `REWIND` cannot be used in asynchronous io.

**Resolution:**

`REWIND` has been modified to work with `BUFFERIN/OUT` unit numbers.

X-21464

PR-22246

The following silently return wrong answers:

- `besyn` with code compiled `-p8, -pd8, -cfc`
- any quad precision `bessel` function (`dbes[sy][01n]` compiled `-cfc, -p8`)
- `dbes[jy]n` `-pd8`

**Resolution:**

The `bessel` functions `besyn` and `besjn` have been fixed to work in all compiler modes. The `bessel` functions `dbesyn` and `dbesjn` have been fixed to work in `-pd8` mode. None of the double precision `bessel` function are supported with the `fc` options `-cfc` and `-p8`.

X-21863

PR-22744

User requests the ability of the compiler to allow string concatenation of a variable into itself. For example,

```
string = 'aa'//string(1:10)//'bb'
```

where `string` is defined as 14 characters and `string(1:10)` is assigned `'0123456789'` would yield `'aa0123456789bb'` instead of garbage.

**Resolution:**

This enhancement has been granted.

X-21864

PR-22748

Given the following source code, the compiler produces an executable that produces a 64 (not 32) character result.

```
real*16 r
r = 1.0
write(*,'(z32.32)') r
stop
end
```

**Resolution:**

This problem has been fixed.

X-22240

PR-23225

The math intrinsic power functions need to special-case the following:  $x^n$  where  $n$  is a scalar integer and  $n$  is equal to 1.

**Resolution:**

This has been granted.

X-22298  
PR-23328

When an already open file is reopened (on the same unit) without an intervening `CLOSE` statement, and a different associated variable is specified, the *original* associated variable is still updated when future I/O occurs, not the new associated variable.

Resolution:  
The FORTRAN `OPEN` statement has been changed to fix this problem.

X-22301  
PR-23334

If either the imaginary or real part of a `complex*8` variable is Reserved Operand (or IEEE Inf/NaN) and you try to write it, the program aborts with a floating point exception (reserved operand).

Resolution:  
The FORTRAN I/O routines that deal with complex numbers have been fixed to handle reserved operand (and/or IEEE NaN/Inf).

X-22338  
PR-23381

According to the *CONVEX FORTRAN Guide*, the `endfile` statement writes an `endfile` record to the file and positions the file after after the `endfile` record. No further records can be read or written without using backspace or rewind. `Endfile` cannot be used on a direct access file. So much for theory : in actuality, `endfile` doesn't seem to do anything. If you compile and run the following, `iostat` after the 'illegal' write is 0. If you open the file with `access='direct'`, it has no problems.

Resolution:  
The above problem has been fixed.

X-22638  
PR-23471

In Cray FORTRAN, a call to `SETPOS` specifying position zero (begin of file) is equivalent to a `REWIND` statement. CONVEX FORTRAN does not allow other FORTRAN I/O statement to be used on the same unit as `BUFFER IN / BUFFER OUT`, so trying to rewind a file in this manner fails. It has been requested that this restriction be removed.

Resolution:  
Rewind will now work with unit numbers that have been used for Cray `BUFFER IN / BUFFER OUT` I/O.

X-22647  
PR-23472

If any of the Cray routines `GETPOS`, `SETPOS`, `UNIT`, or `LENGTH` are called before an actual data transfer has taken place via a `BUFFERIN/OUT` statement, the program fails with a `SIGIOT` error and dumps core. In Cray FORTRAN, this type of coding is common practice and thus does not produce a fatal error.

Resolution:  
`GETPOS`, `SETPOS`, `LENGTH` and `UNIT` will no longer cause a program abort error when used before `BUFFERIN/OUT` statements. They will now return a -1 of the appropriate type when used before any `BUFFERIN/OUT` statements.

X-22672  
PR-23692

The FORTRAN functions `ffrac` and `flmax` do not generate correct results when compiled with the `-cfc` and `-fi` options.

**Resolution:**

Both `flmax` and `ffrac` have been fixed to generate correct answers when used in a program that is compiled with `-cfc` and `-fi` options.

X-22697  
PR-23738

A program hangs instead of creating a core dump. This occurs when intrinsic error trapping is turned on and the variable in question is printed prior to being used in a `SQRT` intrinsic.

**Resolution:**

This problem has been fixed.

X-22789  
PR-23698

While attempting to validate LAPACK, several Floating Point Exceptions were encountered that other computer vendors were not experiencing. These problems have been traced to non-robustness in the complex square root routines for both `complex*8` and `complex*16`.

```

complex*8 x
real*4 t
t = '00800000'x
x = cmplx(0.0e0,t)
print *,x
x = csqrt(x)
print *,x
end

```

**Resolution:**

The complex square root routines have been rewritten to fix the above problem.

X-22925  
PR-24009

`isatty` returns `TRUE` for an unconnected FORTRAN I/O unit.

**Resolution:**

The `isatty` function has been fixed to return `TRUE` only when the I/O unit is connected to a terminal. In all other cases it will return `FALSE`.

X-22927  
PR-24010

If a piece of FORTRAN code compiled `-pd8` contains calls to both `flmax` and `dflmax`, it will not link : you get the following :

```

ld: Multiply defined symbol(s) found:
ld: Symbol '_flmax_' is multiply defined in files '/usr/lib/libU77pd8.a(range.o)'
and '/usr/lib/libU77pd8.a(u_cmx.o)'
ld: ERROR - load aborted due to multiply defined symbols.
ld: No executable produced.

```

Resolution:  
This has been fixed.

X-23028  
PR-24080

User requests for FORTRAN programmers a function similar to `for$getfp` that returns a file descriptor associated with a FORTRAN unit number, rather than the file pointer which `for$getfp` returns. He suggests the name `for$getfd` and is fully aware of the workaround of using `for$getfp` and `fileno`.

Resolution:  
The function `for$getfd` has been added to `libI77.a`. It is documented in the `getfp(3f)` man page.

X-23268  
PR-24405

The vector quad precision compare routines for `.gt.` and `.ge.` do not generate correct answers when compared to zero in all cases (compiled the example at `-O2`).

Resolution:  
The vector quad precision routines for `.ge.` and `.gt.` have been fixed.

X-23295  
PR-24441

The `mvbits(3f)` routines do not work correct in the following modes: `-cfc`, `-pd8` and `-p8`. The `mvbits(3f)` routines should work on a 64 bit integer and not a 32 bit integer in these modes.

Resolution:  
The `mvbits` subroutine has been fixed to work correctly with the `-cfc`, `-p8`, and `-pd8 fc` options.

X-23595  
PR-24740

The size of a Cray binary, unblocked, sequential access file differs from the size of a CONVEX-created file that is intended to be a mirror image.

Resolution:  
This problem has been corrected.

X-23596  
PR-24740

A program containing unblocked Cray binary I/O (CrayUB) and formatted I/O dies with the following run-time error message:

```
sfe: [143] recursive input/output operation
logical unit 35, named '/mnt/lang/noname/TEST/24740/fort.35'
lately: writing sequential unformatted external IO
```

```

.
.
.
IOT trap (core dumped)
```

Resolution:  
This problem has been fixed.

X-23610  
PR-24749

Using an environment variable to set the data format for a Cray unblocked binary file causes the executable to core dump. Specifying the data format in the open statement allows the program to run to completion.

Resolution:  
This has been fixed.

X-23627  
PR-24787

The `ran` function does not appear to be promoted under `-cfc`, `-pd8`, or `-p8` because the return value is always 0.

Resolution:  
The `ran` function has been fixed to expect an `INTEGER*8` argument when linked with any of the following options `-cfc`, `-pd8` and `-p8`.

X-23690  
PR-24881  
PR-25158

The timing functions `dtime` and `etime` are missing from the `libU77pd8.a` and `libU77p8.a` libraries. As a result, when compiling with `-p8` or `-pd8`, the 4 byte routines are pulled in and erroneous results obtained.

Resolution:  
The functions `dtime` and `etime` have been added to the `libU77p8.a` and the `libU77pd8.a` libraries.

X-23691  
PR-24882

There is no 8-byte version of `exit` in `libU77pd8.a` or `libU77p8.a`, causing the 4-byte version to be loaded when compiling or loading with `-pd8` or `-p8`. This causes an incorrect program status to be generated.

Resolution:  
A copy of `exit` has been added to both the `libU77p8.a` and `libU77pd8.a` libraries that correctly handles arguments that are `INTEGER*8`.

X-23783  
PR-24986

The following non-standard program prints one line of output, generates a segmentation violation, and then hangs while aborting. The process continues to consume CPU time until it is killed.

```

integer a(1),i
10 continue
    i = i + 100000
    print *,a(i)
    go to 10
end

```

Resolution:  
This problem has been fixed.

X-24178  
PR-25458

Cannot use any of the F90 intrinsics in IEEE mode because f90real16.o is a native mode object file.

Resolution:  
The build process has been changed to fixed this problem.

X-24553  
PR-25927

The following program, when loaded with the ALL field test libraries, does not generate correct results.

```

program test
character *50 string
100 format ("Happy, happy, joy, joy, this is line ", i2)
open (unit = 10, file = 'stimp'y')
do i = 1, 10
    write (10, 100) i
    call flush (10)
end do
close (10)
end

```

Resolution:  
This problem has been fixed.

X-24751  
PR-26100

The libcfc.a version of etime is only returning a 32-bit result.

```

program main
real a(100000), tarray(2)
e = etime(tarray)
do i = 1,100000
    a(i) = i * 500
enddo
e= etime(tarray)
print *, 'tarray(1) = ', tarray(1)
print *, 'tarray(2) = ', tarray(2)
write (*, '(z16.16)') e
end

```

Resolution:  
This problem has been fixed.

## Lint libraries

X-24043  
PR-25222

The return value for `getpid` in `/usr/lib/lint/l1ib-1c_old` is `int`. However, the declaration in `/usr/include/unistd.h` has it as type `pid_t (short)`. This causes lint to issue inconsistency messages.

**Resolution:**

All of the functions that are declared in `<unistd.h>` that are in `l1ib-1c_old` have been changed to have matching declarations.

X-24141  
PR-25376

Lint complains that `memcpy` returns no value, however, perusal of the C standard, source, and man page confirms that it does return a value. This occurs when using `-ext`, `-str`, and `-std`.

**Resolution:**

Most of the `mem*` routines were incorrectly defined in the appropriate lint libraries. They have been redefined to fix the above problem.

## Man pages

X-22446  
PR-23490

The `bitmil(3bit)` description of the macro `MVBITS` does not list any range limitations on its arguments, unlike the `mvbits(3f)` man page.

**Resolution:**

The range information has been added to the `bitmil(3bit)` man page.

X-23342  
PR-24453

The man page for `errtrap(3f)` has mixed quotes under the value column.

**Resolution:**

The man page has been changed to use the same quotes.

X-23930  
PR-25158

The `etime(3f)` man page needs to indicate the difference between `dtime` and `etime`. Also 'return elapsed execution time' is misleading users into thinking of wall-clock time. It should be clarified that the functions return a process's total elapsed CPU time (user time + system time).

**Resolution:**

The man page has been changed with these corrections.

X-23011  
PR-24081

There is no 3f man page for `for$getfp`.

**Resolution:**

The `for$getfp` function is now documented in `getfp(3f)`.

X-22050  
PR-22979

When standard input, output, and error are redirected, the `getlog` routine returns a null character string for the user's login name. This behavior is not intuitive from the `getlog(3f)` man page.

**Resolution:**

The behavior is due to the behavior of the `getlogin` library routine. This behavior will not be changed. A more appropriate description of the issue was added to the man page.

X-22891  
PR-23951  
PR-24663

The following routines are not documented on the current `topen(3f)` man page: `tstate_lstat`, `tstate_fd`, `tiognoreeot` and `tselectspeed`.

**Resolution:**

`topen(3f)` man page has been updated.

X-22892  
PR-23954

The `topen(3f)` man page needs to document better how EOF's and EOT (physical and logical) are handled with various operations in both synchronous and asynchronous I/O (reads and skips are the biggest problem areas).

**Resolution:**

The `topen(3f)` has been updated to document these issues.

X-23483  
PR-24645

`topen(3f)` doesn't display all the arguments of `tstate`:

```
integer function tstate (tlu, fileno, errf, eoff, eotf,
type,
integer tlu, fileno, dsreg, erreg, resid, size
logical errf, eoff, eotf
```

also, the data type of `type` is not defined.

**Resolution:**

The `topen(3f)` man page has been fixed.

X-24381  
PR-25709

The man page describing `hmalloc`, `libcfc(3f)`, should mention that `hmalloc` calls `malloc(3)` to allocate memory, and `malloc` rounds up the amount of memory allocated to the next highest power of 2. See `malloc(3)`.

**Resolution:**

The libcfc(3f) man page has been updated.

X-24463  
PR-25813

In the synopsis of functions under the topen(3f) man page, the function name `tstate_fd` is misspelled as `tstate_fc`.

**Resolution:**

The man page has been changed to correctly spell `tstate_fd`.

X-24764  
PR-26117

There are too many functions included in libcfc(3f) man page. `Makewhatis` is unable to include all the function names in its database. Message from `makewhatis` is:

```
libcfc.3f: truncating cmdlist from 461 to 300 bytes for DBM's sake
```

The net result is that the command

```
% apropos shiftr
```

doesn't list the libcfc(3f) man page.

**Resolution:**

The libcfc(3f) man page has been split into two man pages: `libcfc(3f)` and `libcfc2(3f)`.

**Installation**

X-24468  
PR-25831  
PR-25835

The install script `load_files` uses the wrong version of perl. It should use `/usr/bin/perl` and not `/usr/local/bin/perl`.

**Resolution:**

The perl script `load_files` has been changed to use `/usr/bin/perl`.

X-24583  
PR-25933

When allowing the installation script to do its own configuration on a machine that doesn't have CXpa, the following conflicting information is generated:

```
Install Configuration: Will not install CXpa libraries (no CXpa)
Install Configuration: Will install CXpa libraries
```

**Resolution:**

The installation script has been fixed to display only:

```
Install Configuration: Will not install CXpa libraries (no CXpa)
```

when CXpa is not installed on the system.

X-25011  
PR-26459

The ALL V1.0 installation script uses an invalid short product name when writing version information to the GIP VDB. The short product name used is "all[as/ld]" and is invalid for the following reasons:

- The short product name must exactly match the regen script for the product found in /usr/lib/gip/regen. The regen script for ALL is "all"
- "all[as/ld]" is an invalid UNIX filename. Required resolution: use "all" as the short product name in all installation scripts.

**Resolution:**

The ALL installation script has been fixed to use the short name of the product as "all."



## Known problems

### FORTRAN libraries

X-14000  
PR-13387

The ANSI FORTRAN standard says that for sequential access: "On output, a new record is created and becomes the last record of the file." [page 12-14, paragraph 12.9.4.1]. However, when writing sequential access tape records, a backspace or rewind which has been preceded with a write, does not write an EOF to the tape. This can leave the tape without an EOF if the tape is subsequently dismounted as part of a multi-volume job.

X-24048  
PR-24935

If FORTRAN I/O is made to a named pipe all works well if the amount of data is greater than the buffer size (4K bytes). If less data are transferred to the named pipe, then the FORTRAN program finishes although no data are read from the pipe.

X-24882  
PR-26302

The following 3f routines will not work with large files: `fgetc`, `fputc`, `fseek`, `getc`, and `putc`. These routines need to call `fseek64` instead of `fseek`.

### Compiler language utility guide

X-23740  
PR-24926

The assembler will treat the digits 8 and 9 in octal constants as the values 010 and 011, respectively. This is not documented in the *CONVEX Compiler Utilities User's Guide*.

X-23438  
PR-24585

Header files shown in the figures do not appear in the index.