Release Notice
CONVEX CXpa V2.0

Document No. 710-008130-006

March 1993

CONVEX Computer Corporation
Richardson, Texas USA

Release Notice, CONVEX CXpa V2.0

Printed in the United States of America

1

Release Notice

Introduction

This document describes the V2.0 production release of the CONVEX Performance Analyzer, CXpa. It highlights new functionality and describes known problems in CXpa. Of special interest in this release is the improved quality of the product. A large effort was made to make CXpa a reliable and more robust profiler. Please refer to this release notice before reporting questions or problems with CXpa; your questions may be answered here. All outstanding bugs are listed at the end of this document. Please review this list to avoid rediscovering known problems.

The remaining sections in this chapter describe the contents of this release.

- Contents of this distribution.
- Enhancements and changes to CXpa.
- Notes and warnings about the product.
- Interaction with CXwindows.
- Known software problems.

For instructions on installing CXpa, see *Installation Procedure, CONVEX CXpa V2.0.*

Contents of This Distribution

The distribution package for this release of CONVEX CXpa consists of:

- This Release Notice
- CXpa Installation Procedure
- CONVEX CXpa User's Guide V1.0
- CONVEX CXpa Reference V1.0
- CONVEX CXpa Quick Reference V1.0
- Distribution media for the software

The specific contents of the software distribution is described in the following tables:

Table 1-1: Software for USA and International Distribution

Qty	Part Number	Description
1	710-005715-008	CONVEX CXpa, V2.0

Table 1-2: Documentation Release Package

Qty	Order Number	Part Number	Description
1	N/A	710-008030-006	Installation Procedure, CONVEX CXpa V2.0
1	N/A	710-008130-006	Release Notice, CONVEX CXpa V2.0
1	DSW-251	710-007230-005	CONVEX CXpa User's Guide V1.0
1	DSW-253	710-004730-004	CONVEX CXpa Reference V1.0
5	DSW-252	710-007330-002	CONVEX CXpa Quick Reference V1.0

Requirements

For the Domestic distribution, this release requires installation of the following:

— CONVEX FORTRAN V8.0, or CONVEX C V5.0. These compilers generate instrumented code for use with CXpa.
— ConvexOS V10.0 or greater.
— CONVEX Assembler, Loader, and Libraries (ALL) V2.0.1.
— To use the X window system features of CXpa, you do not need CONVEX CXwindows. See the section *Notes and Cautions* for further details.

Enhancements

CXpa V2.0 is the first widely distributed profiler release since CXpa V1.3. Listed below are the enhancements since CXpa V1.3.

- **Mflop reporting** — The Loop Performance report now includes a table of performance data for vectorized loops. This table includes metrics such as the loop chime count and performance measurements such as estimated Mflops. Specifically the new report includes the source line number of the loop, nesting level of the loop, vector spill count, vector flops, chime counts, and estimated Mflops. See the online documentation or Appendix A of the *CXpa User's Guide* for more information on the Vectorized loop section of the Loop Performance Analysis table.

- **Graphical user interface** — An X Window interface has been added to CXpa V2.0. This interface allows the user to use OSF/Motif windows to set monitor points, to run programs, and to view profiling information. The GUI for CXpa V2.0 includes a main CXpa window, a command window, a bar graph window, and an extensive online help system. For more information on the GUI, please refer to the online help or the *CXpa User's Guide*.

- **Compiler, CXpa, and library bug fixes** — This release includes a number of bug fixes for both CXpa and the compiler, providing a profiling environment of much higher quality. A number of corrections have been made to the instrumented libraries, including the VECLIB and FORTRAN libraries.

- **Improved Performance Reports** — The performance reports generated by CXpa include more information. The reports and tables have been enhanced to be more readable and succinct.

New commands

The following commands have been added to CXpa V2.0.

- `set pdf` — Sets the name of the performance data file (PDF). If a PDF name is not specified using this command, CXpa uses the default PDF name of cxpa.pdf.

- `pause` — Pauses profiling of a running program and stops execution of the program. This command does not affect the profile time. Once the process has been paused, the `analyze` command may be used to view the profile data that has been collected. In the CRT mode, the command is issued by using the **CTRL-c** key combination.

- `continue` — Continues the profiling of a paused program. When you continue profiling, the monitored program resumes execution, and CXpa resumes collection of profile data.

- `stop` — Stops profile data collection, saves the data collected, and terminates the process being profiled. The `stop` command cannot be used until the process being profiled is paused with the `pause` command. After profiling is stopped, the data collected may be viewed using the `analyze` command.

- `about cxpa` — Displays useful information about the product. The information includes the product code, release date, version number and copyright information. This information is useful when submitting problems or enhancement requests to the Technical Assistance Center.

Changes to existing commands

The following changes have been made to CXpa V2.0.

- **Command line** — The command line interface will no longer use Maryland Windows. Output control will now be managed by the pager identified by the user's environment variable `$PAGER`. If the variable does not exist, output will be piped through the `more` utility.

- `list` — The `list` command no longer supports the display of ordinary ascii text files. Therefore it is no longer necessary to toggle between the "current" ascii file and the performance data file.

- `status` — The `status` command has been renamed to `info` and has been expanded to provide more information. The `info path` command displays the directories appearing in the search path. The information displayed by the `info cxpa` command includes the name and creation date of the executable being profiled, the name and creation date of the PDF, the current list file and current working directory.

- `use` — The `use` command has been renamed to the `path` command. The `path <dir[s]>` command replaces the search path with a new set of directories. The `add path <dir>` command adds directories to the search path.

- `run` — The `run` command will cause CXpa to overwrite an existing profile data file of the same name without prompting the user.
- `profile` — The `profile` command has been replaced by the `set pdf` command.
- **Default Names** — The PDF file name that CXpa uses by default has changed from `cpa.pdf` to `cxpa.pdf`. CXpa will also assume a default executable of the name `a.out`. Finally, the default init file name has changed from `.cpainit` to `.cxpainit`.

Deletion of old commands

The following deletions have been made to the CXpa V2.0 command set.

- `file` — The `file` command has been removed.
- `list file` — The `list file` command has been removed.

Notes and Cautions

This section contains general notes and cautions about CXpa.

- **Patch libI77.a library** — CXpa V2.0 installs a patch release of the libI77.a library to correct instrumentation problems with the previous version. CXpa installs version 2.0.2 of the library in the /usr/lib/palib/ directory. If you have problems profiling FORTRAN programs that perform floating point reading, make sure version 2.0.2 of the library was correctly installed during installation of CXpa. You can determine the version of the library using the `vers` command.
- **Interaction with CXwindows** — While it is preferable that CXwindows be in place for use with CXpa, CXwindows is no longer required. The CXpa installation installs a version of *xterm* on those systems without CXwindows. The *xterm* and the app-defaults file for CXpa allow you to run CXpa in X mode on any X server. You need to obtain a version of the XKeysymDB, which is available from most any X server. This file contains the default key bindings used by OSF/Motif and must be available before starting CXpa with the GUI interface.
- **Serial number** — CXpa is stamped with a unique serial number. At runtime, the profiler checks the serial number of the machine on which it is running. If the serial number does not match the expected one, a user error message is given and execution is aborted.

Known Software Problems

This section contains the known problems with CONVEX CXpa software and compiler instrumentation as of this release notice. Any problems reported after this date may not be reflected in this document. Please refer to this list before reporting a problem to ensure that it has not been reported previously.

- Processes which call `fork()` may not correctly profile under CXpa. When profiling a process that makes a `fork()` system call, the process may not run to completion.
- CXpa cannot profile programs that do not have exit instrumentation. This problem will arise when there is a compiler detectable infinite loop in the control flow to the return from the routine.
- You cannot refer to FORTRAN routine names using uppercase in CXpa commands. When referring to FORTRAN routine names, use lowercase (regardless of how the routine is capitalized in the program itself).
- CXpa cannot run under some ASCII terminals. In particular, the vt220 series of terminals cannot be used to run CXpa in CRT mode. If CXpa cannot run on the terminal you are using, CXpa prints an error message and exits.

A Priority Bugs

- X-27971

Stdout and stderr are really messed up when they get to the tty interface. They seem to be missing buffer flushing and carrage returns. See comment section for an example.

[See also 27710, which has been merged with this record -- brooks]

- X-28303

CXpa dumps core when profiling an application w/fixed scheduling on an 8-headed 3800. Note that it seems to work when I run the application *without* fixed scheduling. At least it worked two times. An offending executable, mmunge-parallel.x, can be found in either gaas:~gbrooks/demo or pixel:~gbrooks/cxpa/demo. To reproduce make sure that mmunge-parallel is compiled -O3, startup cxpa -f, monitor all and then run. When the application gets to the end of it's run cxpa will coredump.

- X-28367

The pager in cxpa isn't working 100%. If you analyzes a run, then q in the middle of the analysis, cxpa thinks it should begin at the line after the command was entered, not at the bottom of the page.

[See comments. -- brooks]

B Priority Bugs

- X-14165

  CXpa can't handle loops with latchbacks or exits implemented by the compiler via branch tables. new instrumentation for such latchbacks/exits needed.

- X-15568

  When a small Ada program is profiled under Cxpa V1.1, cpa seems to hang. The program was compiled with Ada V2.0.1.2 with no optimization.

- X-19224

  The test case PVTs/pvt_06v (and others) fails with either C V4.0 or C V4.1. Sometimes cxpa causes the target program to coredump and somtimes to go off into an infinite loop.

- X-19979

  CXpa analyzes a series of loops incorrectly and consequently reports bogus results.

- X-20898

  CXpa hangs when a program that accesses PVT information is run under it. The program in question calls fork. I've never seen a program run with a fork in it, but apparently it worked at one time.

- X-21772

  CXpa issues the following message when a user routine is compiled at cc4.1 -pa -O0 (or a higher opt):

  Type 'help' for help.
  Reading executable a.out...
  Routine exec at 0x800014ec has no exit instrumentation

  Because CXpa fails to find exit instrumentation for the routine, no profiling is performed for it. This also occurs using -pcc -O0. If the routine is compiled at -no, then exit instrumentation is found and profiling info collected for the routine.

- X-22614

  CXpa cannot be used to profile CXdb. CXpa dies somewhere in analysis. See comments for backtrace. Offending version of cxdb in pixel:/toolmaster/devtools/work2/cxpabugs in a subdirectory named with this X-record number (whatever that turns out to be).

- X-23043

  An application running under cxpa dumps core on a four headed machine, but not on a two headed machine. See comments for extensive details.

- X-24347

  When the piece of code in the comments section is built with the -st and -pa flags, it builds an executible that will raise an exception after execution in cxpa. The executible executes correctly outside of cxpa.

- X-26356

  Ada allows the user to create functions with names such as "+", "-", "*", "**", etc.. However, within CXpa if you want to monitor any of these specific routines currently the only method is via 'monitor all'.

  You cannot say:

  cpa> monitor routine math_op."+"

- X-27287

  If you pause and continue a process a lot, the buttons can get confused. When the process I was running finished(state said finished), the only buttons that were sensitized were pause, stop, close, and help. To get to be able to run again, I had to use rerun. Run woudl not work because of buttons state.

- X-27351

  Run a process and pause it. Then go in and try to set the pdf(foo.pdf). You get 1 messages that say "Cannot open Profile Data File foo.pdf", 2 messages that say "Read failure, could not complete read request", and 1 message "Cannot read Profile Data File header". (problem #1) Then hit continue for the process. Core dumps with IOT trap.

- X-27421

  The "remove" option on change search path doesn't work. If I select a path in bos and hit remove, it goes away out of the box but is in the selection field. Then I hit ok to get rid of the box. The "OK" causes it to be appended again.(design decision after user testing). This I cannot remove a path from the path list. The only way to make it work is to backspace over the name once you get it in the selection field but that is not acceptable.

- X-27641

  When using cxpa, sometimes you can start getting the messages that cxpa.pdf

is in use. Once this happens you have to quit and restart to be able to do anything.

- X-27643

Apparently, whitespace is now significant in evaluating commands in initialization and command files. For instance, "quit " is different than "quit". This may cause existing command files to fail. This didn't happen in CXpa 1.3.

- X-27737

Error messages from the parser in the GUI mode come out as two seperate dialogs. This is confusing and needs to be fixed.

- X-27797

The C test cc/plumhall/conform/lib_d4_6 fails when linked -pa.

I believe this is somehow releated to the cxpa monitor routines that contain their own version of setjmp/longjmp.

- X-27829

Some vectorized loops fail to be reported at all in the analysis. You can do a monitor loop all and then a list monitors and see that they are enabled, but they are not in the analysis at all. See comment for more detail.

- X-27851

If you provide cxpa with file redirection from a file that doesn't exist and into that same file - it will exit with
can't read thread data: errno 14 address 0x11b3
See example for more detail

- X-27927

The nas kernal benchmark will consistantly core dump cxpa at O3, but will run outside of cxpa. Try:
monitor copy
monitor mxmtst
monitor mxm
monitor ffttst
monitor cfft2d1
monitor cfft2d2
monitor chotst
monitor cholsky
monitor btrtst
monitor btrix
monitor gmttst
monitor gmtry

```
monitor emitst
monitor emit
monitor vpetst
monitor vpenta
run < cpa.input
```

Where cpa.input is an empty file.

- X-28140

If you startup cxpa with a display argument, the xterm that is created during the run will use the shell display enviroment. Try

```
cxpa -display alternateSun:0 a.out &
mon all
run
(now the xterm will appear on the the shells $DISPLAY.)
```

- X-28207

I was trying to use the Search All Text feature
in the CXpa help window (IDTK) to search all the help pages
for the string " file-name ". However it found matches for
filename rather than file-name.

- X-28386

There are several places in the GUI where lists of routines come up. While the routines themselves are correct, they are not consistant in there ordering. Without this consistancy there is currenty no way to automate the selection of a particular routine. The tests also cannot use these windows in any type of screen capture. Windows that I have found so far that show this behavior are the analyze (loop, pregion) windows.

C Priority Bugs

- X-14298

CXpa reports unreasonably large routine times for an executable that raises an exception and was compiled with -st.

- X-15725

The CXpa commands "file" and "list" are unable to find an Ada source file that is in the current directory.

- X-16422

When profiling C code with instrumented libraries, CXpa generates extra calls to ap$flsbuf when program IO has been redirected to a file.

- X-19221

CXpa produces extremely large timings for a Fortran code that contains many READ statements. This problem occurs on a 8k/1k file system, but _NOT_ on a 64k/8k file system.

- X-20253

CXpa's status command tries to find all monitorable source files, even if nothing has yet been selected for monitoring. This results in error messages for all library source files because they aren't found in the search path.

- X-20985

CXpa produces incorrect numbers for monitored routine MAIN_.

CPU Time (less children)	CPU Time (plus children)	Times Called	Routine name
18446744073709.551	2661867897944539000.0%	18446744073709.551	266186789794 4539000.0% 1 MAIN

- X-23044

CXpa is reporting loop optimization information that conflicts with opt report. I would tend to believe opt report in this case. See comments for code, cxpa output and opt report.

- X-24028

CXpa reports that the included loop has a nested exit - which either needs more explination or is incorrect.

Also, note that the average iterations for the loop in question is more than the max for the loop.

Please see comments for more detail

- X-24068

CXpa returns bogus percentages after an exception is raised and processed. See detail section for the source.

Routine Performance Analysis
(sorted by CPU time (less children))

CPU Time	CPU Time	Times

```
(less children)        (plus children)        Called  Routine name
 0.191m  2387.5%        0.191m  2387.5%            1  except2.inner
 0.012m  150.0%         0.203m  2537.5%            1  except2
```

- X-24069

  CXpa reports bogus cpu times when an exception is raised and not handled.

```
                                        Routine CPU Time Analysis
                                    (sorted by CPU time (less children))
 CPU Time Analysis          CPU Time Analysis
  (less children)            (plus children)
Min.     Max.     Avg.     Min.     Max.     Avg.   Routine name

9223372036854.775    0.000    0.000 9223372036854.775    0.000    0.00092233720368547.775    0.000    0.000 9223
```

- X-24072

  CXpa returns a bogus cpu time for a loop that has been interchanged. The only thing that I see different here is that the loop has a pragma vector_unit. I think that I have seen loops like this work correctly before.

```
                        Loop Transformation Performance Analysis
------  ---------   -------   ------ ------ ------   ---------------
  90    0:SM,Hs           4      128    128    128         0.000022
  89    1:S,I             4       16     48     36    18446744073709.55078
```

- X-24348

  cxpa sometimes returns very large numbers for cpu times. Below is a list of sources that you can compile at the given opt. level to see this behavior. The sources live in /swenv/twork/cxpa/ada/tasking/acvc

Source	Opt. Level
c93006a.a	-O2 -O3
c97205a.a	-O2 -O3
c97205a.a	-O2 -O3
c97205b.a	-O2 -O3
c97305a.a	-O2 -O3
c97305b.a	-O2 -O3
c97305c.a	-O2 -O3
c97305d.a	-O2 -O3

- X-24365

  if the assembly file given in the comments section is fed to

cxpa it will never exit. To reproduce this behavior:
as <file>
cc -pa <file>.o
cpa a.out
run <-- this run will finish
monitor all
run <-- this run will not finish

Another thing is that cpa is chomping on the cpu while the executible is asleep.

- X-25017

CXpa does not have a regen script in the GIP source tree (and therefore in /usr/lib/gip/regen).

- X-25101

CXpa will exit with a SIGBUS if you set your home shell variable to '.'

593-pixel--> set home = .
594-pixel--> cpa foo

CONVEX Performance Analyzer

Type 'help' for help.
Reading executable da003.e-O3-pa...

cpa exiting with signal SIGBUS (bus error).

595-pixel--> vers 'which cpa'
/usr/convex/cpa: 1.3

- X-25102

CXpa returns bogus results when 'monitor routine all' is used in the ada program included in the comments.

- X-25295

CXpa exits with a bus error when analyzing a pdf file.

- X-25297

CXpa fails to return percentage values even though it has reasonable cpu times for all routines.

- X-25515

The list routine command does not seem to work properly with uppercase routine names. For example, I have a routine named SUB1 in my source code. However, when I try to perform "list SUB1", it can't find it:

```
(cpa) list SUB1
    No listable routine named SUB1.
```

However, if I type SUB1 in lower case, it seems to work:

```
(cpa) list sub1
 r   33      SUBROUTINE SUB1(BOGUS, BOGUS3)
     34      INTEGER MATRIX(5,5), BOGUS, BOGUS2, BOGUS3, FACTOR
```

This seems like a bug to me.

- X-25937

A program with "too many" Fortran formatted READ statements causes CXpa to yield extremely large percentages for for$x_rnew. Up to 2878 READ statements give a reasonable routine analysis, but when there are 2879 reads or more, the CXpa table overflows. The cut-off point between good and bad CXpa runs is different on different C2 machines--on serial number 8500, it is 1489.
On hydra, this problem was not reproduceable even with as many as 15000 READ statements.

- X-26824

Cxpa does not work well with generic tasks. When using special orders of generics and tasking, cxpa causes some tasks to die causing tasking errors. The problem submitted here takes about 4 minutes to create the tasking errors. Please read comment for further details.

- X-27142

The parser in cxpa dosn't seem to catch ' characters in the command line interface (tty mode). try the command 'run''

- X-27248

In a parallel analysis, if a parallel region is a result of a distributed region, it needs to flaged with the same letter as it would in the loop analysis section.

Please see comments section for more detail.

- X-27255

The first time the help window comes up in a session, it is too small and the user will have ot resize to see the screen dumps within the window. A lot of the beginnings of the help windows have screen dumps so most users will end up having to resize.

- X-27265

  Select monitor all, run, then pause the process. Then stop it. If you then try to quit cxpa, it will tell you the process is still paused.

- X-27304

  When cxpa core dumps, it leaves temporary files in /tmp.

- X-27336

  The list monitor command does not show any output when there are no monitor points within the current 10 line range. Look at the example for more info.

  It seems to me that LIST MONITORS should either say that there are no monitor points in the current 10 line source range or (preferably) show the next 10 lines with monitor point, regardless of how far apart they might be in the source.

- X-27360

  In the help text for the Control Center, there are tables with underlined words. This is a little confusing since these are not hyperlinks but just underlining for clarity. If underlining is going to be the hyperlink mechanism, these underlines might need to be deleted.

- X-27367

  Under the "Info" pulldown, clicking on the "Monitor List" takes a long time, then creates this window "Monitor Point Information" that is about 1/4" by 1/2". This is only for a small benchmark.

- X-27725

  When there is an error in the command file, the error message that is output points to (with ^) to the wrong line. See example given with bug.

- X-27806

  Help message text (width) is formatted to some hard-coded length. No matter how wide the GUI Command Window is sized, each line of the help text output is automatically formatted to what appears to be 80 chars.

- X-28099

  When you bring up CXpa from a CRT, you get the following message:

  error: unable to open display

While I realize that one solution is to start cxpa with -nw,
I think that CRT users shouldn't have to do this.

- X-28208

Using the GUI, when you run a program without enabling any monitor points, an information box appears telling you that nothing is being monitored. That's fine. However in the information box, there is a help button. This button just brings up the default help page.

As we have done in other situations, we should remove the help button when there is no specific help page for an info or error message. The user will just get agravated if there is no specific help.

- X-28231

When running cxpa on cxdb - the analysis reports generated sometimes have large numbers in them.

- X-28246

Resizing CXpa's main window will cause the online help window to resize.

- X-28247

With large fonts, like -*-courier-bold-r-*-*-24-*-*-*-m-*-*-*, the text in many of CXpa's buttons, is not all visible. Not a big deal, but it looks bad.

- X-28248

In the online help window, the search box area is too large and partially occludes some of the surrounding buttons. This is particularly noticable with a larger default font like, -*-courier-bold-r-*-*-24-*-*-*-m-*-*-*.

- X-28249

When you start CXpa up in the foreground (in GUI mode), you cannot then ^Z the shell to put it background. You should be able to ^Z it.

- X-28358

Changing the search path several times while also doing an 'info cxpa' confuses the list file entry in the info cxpa output. See comment for example.

- X-28378

The set pdf box should not let you set the pdf file to an invalid name like ' '. If you do this, then the analysis window is tied to the filename ' ' because it will not let you analyze anything. The run dialog is still tied to the default cxpa.pdf and will let you run. (This is similar to bug x27253 for testing)

There are 4 dialogs that come up and tell you that the pdf file

- X-28393

When you quit with a paused process, cxpa asks you if you want to kill the process. If you answer y or just hit return you get two messages. One tells you that it killed the process, the other tells you that it couldn't kill it.

- X-28394

If you attempt to quit from cxpa with a paused executable, you will see the following message:

(execuatble name) is paused, quit CXpa anyway ([y]/n)?

if you type any character other than a n, cxpa will terminate as if you hit a 'y'.

- X-28425

The monitor point information dialog box has some strange characters when you display the MAIN_ routine of a FORTRAN program compiled with -pab. The source code is in a comment.

The executable is in /swenv/twork/new_cxpa/CC/block.out

D Priority Bugs

- X-21071

CXpa should be improved to either report that iteration counts are invalid or to calculate them in the case where a loop nest contains an exit. As it works now, it prints invalid numbers here.(average iteration count > max count)

- X-26926

It would be nice if typing 'close' in the command window would close the window.