Release Notice
CONVEX Consultant V8.2

Document No. 710-009530-000

April 1990

CONVEX Computer Corporation
Richardson, Texas USA

Release Notice, CONVEX Consultant V8.2

Printed in the United States of America

Table of Contents

List of Tables

1

Release Notice

1.1 Introduction

This document describes the V8.2 production release of the CONVEX Consultant. It supplements the current documentation with information and features that were developed too late to be included in the Consultant documentation. Always refer to this release notice before reporting questions about or problems with the Consultant. Answers to your questions may be in this document, which lists fixes and workarounds to help you avoid rediscovering known problems.

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 product.
- Section 4 describes enhancements to previous functionality.
- Section 5 describes fixes to the software.
- Section 6 describes known software problems.
- Section 7 contains fixes to the documentation.
- Section 8 lists known documentation problems.

The CONVEX Consultant is a set of useful utilities for developing CONVEX software applications, namely, *bprof*(1), *csd*(1), *gprof*(1), *prof*(1), and *pmd*(1). Included in the Consultant release are the following utilities:

- *bprof*(1)— statement level execution profiler.
- *csd*(1)— symbolic debugger.
- *gprof*(1)— process execution call graph profiler.
- *pmd*(1)— post mortem dump for program aborts.
- *prof*(1)— process profiler.

For instructions on installing the Consultant, see *Installation Procedures, CONVEX Consultant V8.2.*

1.2 Contents of This Distribution

The distribution package for this update of the CONVEX Consultant consists of this document and distribution media for the software. If you already have the Consultant, you will receive the Update Package. If you are receiving your first Consultant, you will receive the Release Package. The specific contents of the software distribution are described in the following tables:

Table 1-1: Release Package

ITEM	QTY	TYPE	PART NUMBER	DESCRIPTION	FORMAT
1.	1	Mag.	710-006015-001	CONVEX Consultant, V8.2	Installsw
2.	1	Document	710-009730-000	Installation Procedures, V8.2	
3.	1	Manual	740-002530-203	CONVEX Consultant User's Guide, 8th Ed.	

Table 1-2: Update Package

ITEM	QTY	TYPE	PART NUMBER	DESCRIPTION	FORMAT
1.	1	Mag.	710-006015-001	CONVEX Consultant, V8.2	Installsw
2.	1	Document	710-009730-000	Installation Procedures, V8.2	

1.3 Notes and Warnings

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

- For the Domestic distribution, this release requires the installation of ConvexOS Utilities USA V8.0 and ConvexOS V8.0. For the International distribution, this release requires the installation of ConvexOS Utilities International V8.0 and ConvexOS V8.0.

- In the remainder of this document, the term *CONVEX C* refers to version 4.0 of the CONVEX C compiler. This compiler is executed with the *cc* command. On systems that have the new compiler installed, the compiler executes with the *pcc* command instead of the *cc* command. In the remainder of this document, the term *pcc* refers to the old compiler.

- Some *csd* responses that appear to be caused by Consultant bugs are generated as the result of compiler bugs. In these cases, the compiler does not provide correct information to the Consultant. Preprocessor *#include* commands that contain executable code can disrupt the flow of line number information from the compilers to *csd*. When this occurs, *csd* can give "event point not permitted" messages for lines that actually contain executable code.

  At optimization levels higher than *-no*, *cc* transmits less executable line and nested variable information to *csd*. This can lead to "is not active" messages for variables that really should be accessible and "event point not permitted" messages for lines that actually contain executable code.

  Note the following information for FORTRAN:

  1. The procedure for finding the instruction at which execution aborted may fail. This procedure is outlined in section 3.2.3 of the Consultant User's Guide. The compiler sometimes produces direct object code that differs from that produced by the assembly of a *.s* file produced with the *-S* flag. To avoid this problem, assemble and use the code in the *.s* file. Use the *a.out* file produced by the command

     fc xx.s *.o

  2. Active variables and dummy arguments become inactive after stepping past an ENTRY statement. This problem has been fixed in *CONVEX FORTRAN V6.0*.

Note the following information for C:

1. Code compiled by *CONVEX C 4.0* at optimization levels **-O0** and above gives *csd* little information about the inner scope of variables and executable lines of code. Compile the source file at optimization level **-no** to obtain additional information.

2. *csd* may attribute source lines to the wrong source file in *pcc* compiled code when #include preprocessor commands are nested.

3. For *pcc* compiled code, *csd* reports local external array variables as undefined.

4. *CONVEX C* implements enumerated types as integers. *csd* does not know that they were originally declared as enumerated types. Thus, the response from *csd* for C programs that contain enumerated types depends on whether the program was compiled with *pcc* or *cc*.

These compiler problems may be fixed in the next releases of the *cc*, *fc*, and *vc* compilers. Check the product release notices when they appear.

1.4 Enhancements

1.4.1 bprof

none

1.4.2 csd

1. You can redirect program output to another window if you run *csd* under X Windows. You do this in the *run* command. Use the ConvexOS *tty* command in the other window to get the window name. Then use

   run .... > /dev/windowname

   where "windowname" is ttyXX.

   Using this technique, you can see the output of the program in "windowname" while typing commands unobstructed in the window in which you are running *csd*.

2. A number of changes have been made to *csd* to make it work correctly with CONVEX C V4.0.

3. *csd* now makes void compatible with any basic data type. This really only adds the capability to assign and pass pointer to void as a pointer to any basic type since no CONVEX C compiler will permit a variable of type void.

4. The -C command line switch has been added. When this switch is specified, a function can be referred to by a prefix of its name. For example, if functions **printHex** and **printDecimal** are both in the executable, the command

   stop in print

   causes *csd* to display a list like

1. printHex()
2. printDecimal()
3. printf()

After the list of all function names that begin with the specified prefix has been displayed, *csd* prompts you for the number of the function you want.

5. The hex form for a C character variable is now printed in addition to the character form when you specify "**format hex**".

6. The response to the dump command now includes more information for COMMON data in active FORTRAN subprograms. The module and COMMON name and the values of all variables are now printed for each active COMMON.

7. Non-active FORTRAN COMMON and local variables can now be accessed. To access local variable **index** and **COMMON bar** for FORTRAN subprogram **foo**, use the following command sequence:

```
func foo
print index
print bar
```

8. *csd* now prints all union variations when a "print XX" command is given for a variable XX which is a union. The resulting output closely resembles that of a structure variable. Previously, *csd* printed [**union**] for such a print command.

9. *csd* now includes a *goto* command. The form of the command is

goto lineNumber

The line number must be in the current function. *csd* does not know what the correct values for the environment (for example, sp, fp, PSW). Thus, ensure that your environment is correct.

1.4.3 gprof

none

1.4.4 pmd

The information printed for FORTRAN COMMON data has been improved. For each active COMMON, *pmd* prints the module and COMMON name and the value of each variable in that COMMON.

1.4.5 prof

none

1.5 Software Fixes

The following bugs have been fixed for this release.

1.5.1 bprof

The usage statement from *bprof* now shows that a space is required between the -I switch and the directory pathname argument.

1.5.2 csd

The following *csd* problems have been fixed:

- The limit on the number of lines in any one source file has been raised from 20480 to 51200.
- *csd* now gives fewer incorrect "is not active" replies to requests to print the values of variables.
- *csd* now prints the correct number of hex digits for FORTRAN and C variables.
- Many stack overflow and stack underflow situations have been eliminated.
- The number of elements specified by the last preceding **set num_elements ...** command are now printed for arrays in output for the **dump** command.
- *csd* no longer responds with source for function **main** in FORTRAN programs.
- The size of a variable that can be traced is now commensurate with the size of a variable that can be printed. Previously, traced variables were limited to 2048 bytes. The limit is now 32768 bytes.

1.5.3 gprof

none

1.5.4 pmd

pmd now processes the number of elements specification correctly when printing arrays.

1.5.5 prof

The field in the output listing for number of calls now can contain up to eight digits. In the previous version of *profP, if the number of calls was a million or more, spaces did not separate this field from the field for cumulative seconds data.*

1.6 Known Software Problems

This section describes known problems with CONVEX Consultant software as of March 27, 1990. This document may not reflect problems reported after this date. Please refer to this list before reporting a problem to ensure that it has not been reported. Descriptions of serious problems include known workarounds.

1.6.1 bprof

- *bprof* may fail to produce correct call counts when the *-I* flag is used to collect source information from remote directories and the *-m* and *-f* flags are also used.
- *bprof* provides no early warning message when the operand to the *-I* flag is not a legal directory. The message is produced later trying to refer to the illegal directory.

1.6.2 csd

- *csd* may get an internal error (missing trace id) if processing continues past the exit from MAIN_ and a *when* command is in effect.
- *csd* uses breakpoints of its own to turn on tracing when a routine is entered. For example, a command like

  trace sub.i in sub

  results in a breakpoint in the same place that a

  stop in sub

  command puts one. If you have reached a line like

  call sub()

  and issue a *next* command, *csd* stops for its own breakpoints in *sub* just as it would for one of yours.

1.6.3 gprof

none

1.6.4 pmd

none

1.6.5 prof

none

1.7 Documentation Fixes

The *CONVEX Consultant User's Guide* has not been updated for this release. Users with no prior release will receive the *CONVEX Consultant User's Guide* which was part of the Consultant V8.1 release. These release notes document the new features and the problems that have been fixed in the Consultant since the V8.1 release.

1.7.1 bprof

The manual page for *bprof* now shows that a space is required between the -I switch and the directory pathname argument.

1.7.2 gprof

The manual page for *gprof* now includes a discussion that defines the term "split ticks".

1.8 Known Documentation Problems

The *CONVEX Consultant User's Guide* has not been updated to reflect the enhancements made to the Consultant in Releases 8.1 and 8.2.