

Release Notice
CONVEX Consultant V10.1
Document No. 710-009530-003

July 1992

CONVEX Computer Corporation
Richardson, Texas USA

Release Notice, CONVEX Consultant V10.1

Copyright 1987, 1988, 1989, 1990, 1991, 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.

CONVEX and the CONVEX logo ("C") are registered trademarks of CONVEX
Computer Corporation.

ConvexOS is a trademark of CONVEX Computer Corporation.

Printed in the United States of America

Table of Contents

1 Release Notice	
1.1 Introduction	1-1
1.2 Contents of This Distribution	1-1
1.3 Notes and Warnings	1-2
1.4 Enhancements	1-3
1.5 Software Fixes	1-5
1.6 Known Software Problems	1-6
1.7 Documentation Fixes	1-7
1.8 Known Documentation Problems	1-8

List of Tables

1-1 Release Package	1-2
1-2 Update Package	1-2

Release Notice

1.1 Introduction

This document describes the V10.1 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 work arounds 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 V10.1*.

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.	710-006015-004	CONVEX Consultant, V10.1	Installsw
2.	1	Document	710-009730-003	Installation Procedures, V10.1	
3.	1	Document	710-009530-003	CONVEX Consultant, V10.1 Release Notice	
4.	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.	710-006015-004	CONVEX Consultant, V10.1	Installsw
2.	1	Document	710-009730-003	Installation Procedures, V10.1	
3.	1	Document	710-009530-003	CONVEX Consultant, V10.1 Release Notice	

1.3 Notes and Warnings

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

- The Consultant V10.1 is the final release of this product. *csd* and *pmc* are no longer supported. *prof*, *gprof*, and *bprof* will be continued as part of the *CONVEX CXpa* release.
- For the Domestic distribution, this release requires the installation of ConvexOS Utilities USA V10.1 and ConvexOS V10.1. For the International distribution, this release requires the installation of ConvexOS Utilities International V10.1 and ConvexOS V10.1.
- This Consultant release is being made primarily for customers with systems that have ConvexOS V10.0 installed. This release may not work correctly on systems that have ConvexOS 8.x and 9.0 installed.
- In the remainder of this document, the term *CONVEX C* refers to version 4.1 of the CONVEX C compiler. This compiler is executed with 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 hardware problems.

Note the following information for CONVEX 3800 series machines: some preproduction systems step two instructions instead of one when single stepping starting on a call instruction. The system stops on the second instruction of the called function instead of the first. This may cause *csd* to stop at unexpected source lines when using the *step* or *next* commands.

Other inappropriate responses are 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* and *fc* transmit 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. CONVEX suggests that the new *cxdb* debugger be used to debug optimized 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 CONVEX 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
```

CONVEX suggests that the new *cxdb* debugger be used to more easily determine the instruction that aborted execution.

2. The information that *csd* receives from the compiler for assumed sized arrays is that they have dimension 1. An assumed size array is an array argument to a subroutine for function that is declared with an asterisk as its leftmost dimension. The compiler has no better information to give. When *csd* is asked to print such an array, it prints only the first element. To avoid this situation, declare the array as an adjustable array: one whose dimension(s) are passed to the subroutine for function as arguments. That way, *csd* will know how many elements to print.

Note the following information for C:

1. Code compiled by CONVEX C 4.1 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, or use the new CONVEX *cxdb* debugger that was designed for debugging optimized programs.
2. 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* and *fc*, compilers. Check the product release notices when they appear.

Note the following information for *veclib*:

csd initially begins execution in the mode that causes the child process to stop and interrupt *csd* when thread creation and destruction events occur. For some programs that include *veclib* routines, this results in strange intermittent behavior. The remedy is to have *csd* run the child process in the mode in which no thread trapping is done. Use the command

```
set threads false
```

to put *csd* and the child process in this mode.

1.4 Enhancements

1.4.1 bprof

none

1.4.2 *csd*

1. A new form of the assign command has been added. The old form
assign variable = value
requires that the data types of the variable and the value agree. In some debugging situations, you may want to assign a value that *csd* thinks is the wrong type to a variable. For example, you may need to assign a constant to a pointer. Use the new form

assign variable @ value

to tell *csd* to omit the type check.

2. If you run *csd* under X Windows, you can redirect program output to another window. 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*.

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* that is a union. The resulting output closely resembles that of a structure variable. Previously, *csd* printed [union] for such a print command.

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 installation script

The Consultant version V10.1 installation script now works correctly when the *gut* command has been applied to */vmunix*.

1.5.2 bprof

The following problems have been fixed:

- Inappropriate warning messages for FORTRAN ENTRY statements have been removed.
- *bprof* now prints a message that a needed source file is not available. Prior versions aborted.
- The usage statement from *bprof* now shows that a space is required between the -I switch and the directory pathname argument.

1.5.3 csd

The following *csd* problems have been fixed:

- *csd* now runs on the 3400 and 3800 models of CONVEX computers.
- *csd* now knows how to disassemble all of the new instructions that have been added to recent new machines in the Convex product line.
- *csd* no longer aborts when asked to print a C function argument that points to a C function.
- *csd* no longer aborts with an "internal error - trace id not found" message.

- 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.

1.5.4 gprof

none

1.5.5 pmd

The following problems have been fixed:

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

1.5.6 prof

The field in the output listing for number of calls now can contain up to eight digits. In the previous version of *prof*, 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 July 9, 1992. 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 work arounds.

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* 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* that was part of the Consultant V8.2 release. These release notes document the new features and the problems that have been fixed in the Consultant since the V8.2 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 8.2, 9.0, or 9.1.