

**Release Notice:
GNU C Compiler for
AViion® Systems**

Release 2.5.8.1

August 1994

Part number 085-600422-00

This release notice applies to the following models:

P001
Q001

Restrictions and Trademarks

This software is made **available** solely pursuant to the terms of a DGC license agreement which governs its use.

Restricted Rights Legend:

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at Defense Federal Acquisition Regulation (DFARS) 252.227-7013 and in subparagraphs (a) through (d) of the Commercial Computer Software Restricted Rights clause at Federal Acquisition Regulations (FAR) 52.227-19, whichever may apply.

DATA GENERAL CORPORATION
4400 Computer Drive
Westboro, Massachusetts 01580

AViiON is a U.S. registered trademark of Data General Corporation.

DG/UX is a trademark of Data General Corporation.

UNIX is a U.S. registered trademark of Unix Systems Laboratories, Inc.

Release Notice: GNU C Compiler for AViiON® Systems Release 2.5.8.1

085-600422-00

Revision History:

Effective with:

Original Release – August 1994 DG/UX System 5.4 Release 3.10

Contents

1	Introduction	5
2	Product Description	5
3	Environment	5
4	Enhancements and Changes	6
5	Notes and Warnings	6
5.1	Notes	6
5.2	Warnings	6
6	Documentation	7
7	Software Distribution	7
7.1	Media	7
7.2	Files	7
8	Installation Instructions	7
9	Preparing a Software Trouble Report (STR)	8
9.1	Gathering STR Information	8
9.2	Gathering Materials for a Software Problem	9
9.3	Sending Media	10

1 Introduction

This release notice describes the GNU C that runs under DG/UX. In addition, this notice provides information not currently available in the GNU documentation (such as information developed after the current manuals were printed or corrections to the current manuals).

You can print additional copies of this release notice after you have installed GNU C. The file `/usr/release/gcc_2.5.8.1_rm` is suitable for lineprinters. Please read this release notice completely before installing GNU C. Detailed installation instructions are provided in this release notice.

2 Product Description

GNU C is an ANSI-compliant C compiler that is also compatible with PCC. GNU C supports a high level of optimization, subject to user control through various command-line options. The GNU compiler offers a number of extensions to ANSI C, including the **asm**, **typeof**, and **inline** constructs. GNU C adheres to the 88open Object Compatibility Standard [OCS] in terms of assembler code generated, calling sequence, and alignment. See the "Notes" section for a listing of exceptions to the OCS.

The GNU C compiler is invoked either by `/bin/gcc` or `/bin/cc`. The `/bin/cc` program invokes the GNU C compiler with the `-traditional` switch, and is compatible with System V in terms of its switches. The `/bin/gcc` program supports the normal GNU switches, and defaults to mostly ANSI C plus the GNU C extensions.

Data General's 2.5.8.1 version of GNU C is based on Version 2.5.8 of the compiler from the Free Software Foundation. The GNU General Public License provides you with the opportunity to obtain source code for the GNU C compiler for the AViiON. This is available on the AViiON Contributed Software distribution (Model #R006AZN20A) from Data General. **The GNU General Public License does not in any way restrict the distribution of software compiled with GNU C or linked with glibc or any other libraries supplied with DG/UX on AViiON systems.**

3 Environment

GNU C 2.5.8.1 runs on an AViiON system running DG/UX Release 5.4 Revision 3.10. The GNU C release occupies approximately three megabytes of disk space.

4 Enhancements and Changes

Please note the following enhancements and changes to GNU C for this release:

- **-fPIC** now generates correct optimized code.
- If **-m88110** is specified, then **-muse-div-instruction** and **-mno-check-zero-division** are assumed. (**-mno-use-div-instruction** and **-mcheck-zero-division** are ignored if **-m88110** is specified).
- Warnings are no longer given for floating point constants out of range; they are represented as an IEEE infinity value.

5 Notes and Warnings

5.1 Notes

- The following options apply to both the **/bin/cc** and **/bin/gcc** interfaces to the GNU C compiler. To use any other GNU C option with **/bin/cc**, you should prefix the option with either **-Wp**, for a preprocessor option or **-W0**, for a compiler option, e.g. **-W0,-fno-delayed-branch**.

```

-c
-g
-o outfile
-p
-v
-C
-Dname[=value]
-E
-H
-Idirectory
-O
-S
-Uname

```

- Because the m88110 calling sequence passes arguments in registers, any attempt at handling variable numbers of arguments is likely to meet with limited success unless you use the standard header files (**varargs.h** or **stdarg.h**).
- The location of compiler files in **/usr/lib/gcc** is subject to change in a future revision. Programs should not depend on this location. Instead, they should access the compiler through **/bin/gcc**, **/bin/cc**, or **/lib/cpp**.

5.2 Warnings

- Contrary to the OCS, GNU C invoked by **/bin/gcc** does not put strings into writable memory unless you use either the **-traditional** or the **-fwritable-strings** switch. When invoked via **/bin/cc**, GNU C assumes the **-traditional** switch, and places strings in writable memory.

- You should note that token concatenation works differently under ANSI C (if you use **/bin/gcc** without the **-traditional** switch) as compared to traditional, PCC-compatible C (using either the **-traditional** switch or **/bin/cc**).
- Programs with extremely long initializers take considerable time and swap space to compile.

6 Documentation

The on-line manual *Using and Porting GNU CC*, by Richard M. Stallman of the Free Software Foundation, provides complete information on compiler options, as well as chapters on GNU extensions to the C language and incompatibilities with "traditional" C compilers. This manual is accessible through the **info** system, a menu-driven browsing tool. See the **info(1)** manual page for an introduction to the system.

The manual can also be accessed with the GNU Emacs Info document browsing subsystem (GNU Emacs is distributed with the GNU Toolset, Model R028A).

For general information on the C language, see Harbison and Steele, *C: A Reference Manual* (069-100226). For command-line options to GNU C, consult the manual page **gcc(1)**.

7 Software Distribution

7.1 Media

GNU C accompanies DG/UX.

7.2 Files

The file **/usr/release/gcc_2.5.8.1.fl** lists the files that constitute the GNU C .

8 Installation Instructions

GNU C software is installed during the installation of DG/UX. See the DG/UX release notice and the manual *Installing the DG/UX System* for information on installing DG/UX. Optionally, to add GCC to the GNU Emacs info subsystem, see "Documentation" above.

9 Preparing a Software Trouble Report (STR)

9.1 Gathering STR Information

If you find an error in the GNU C software or the GNU documentation or if you have suggestions, please fill out and return a Data General Software Trouble Report (STR). If your contract permits, you can report the information called for in this section to your Data General representative. To help expedite STR processing, include only one problem or suggestion on each STR form. Please follow these guidelines when filling out your Software Trouble Report.

1. List the name of the product as GNU C on the STR; calling the product gcc may lead to misfiled or delayed STRs. The model number is the same as the DG/UX release, Q001 or P001.
2. Decide what kind of STR you are writing:
 - Enhancement -- describe the proposed enhancement clearly and tell why you want it. The better we understand what you want, the easier it is for us to evaluate your request.
 - Documentation error -- give the page and section or paragraph, and tell why you think there is an error.
 - Software problem -- clearly and specifically state the problem so that support personnel can try to reproduce it. Avoid phrases like "the program does not work" or "fails."
3. On the STR form provide all of the following information:
 - Date
 - Revision of the product. Use the `-v` option to obtain the DG version number of the compiler. A line such as


```
gcc version (88open OCS/BCS, DG-, )
reports version .
```
 - Revision of the operating system
 - Names and revisions of other software this product uses
 - CPU type
 - CPU revision
 - The command line, complete instruction, or program name that caused the problem
 - How often the problem occurs and how serious it is
 - The action(s) necessary to reproduce the problem
4. If the problem occurred soon after installing a new revision of the operating system, GNU C, or new hardware, note this.
5. If you received an error message, please write down the text (and number, if there is one) of the message. Also, note when you received the message(s) (for example, during compiling,

linking, executing, etc.)

6. In the case of a program crash, a core dump file may be produced in the directory from which the program was started. If such a file is found, include a copy of it with the STR.

9.2 Gathering Materials for a Software Problem

When submitting a Software Trouble Report (STR), it is very important that you supply all of the information necessary to recreate the problem you are experiencing:

- If the problem occurred when you attempted to compile your program, you should submit with your STR a tape containing either
 - a. the source file of the program and all `#include` files it used, or
 - b. the output of preprocessing the source and include files (this can be created using “`gcc -E source.c > source.i`” or “`cc -P source.c`”).

You do not need to send the `#include` files supplied with this product, the standard DG/UX system include files in `/usr/include`, or the entire GNU C , as these are archived at Data General.

- If the problem occurred while you were linking or running your program, you should also supply sources for all modules and an exact description of the `ld` command line.
- If the problem occurred while you were executing your program, and the program read input from file(s), you should supply the appropriate input file(s) that reproduce the problem. Also, please indicate what shell (`sh` or `cs`) you used to execute the program and any special conditions (for example, the program fails only when running in shell layers or when using the network).

Please supply any makefiles or shell scripts used to compile, link, and execute your program, so that we can try to recreate the problem.

You can help us find a speedy solution to the problem if you take the time to try to isolate the problem. Use any diagnostics the compiler may supply, the source language debuggers, or your knowledge of what the program is intended to do (or what source lines you recently changed that may have caused the problem to suddenly occur). You may find the following helpful:

- Are you aware of the three main modes of the compiler: `-traditional`, no option, and `-ansi`?
- A useful discussion on "Have you Found a Bug?" may be found on pages 70-71 of *Using and Porting GNU CC*.
- Does the program work correctly with or without optimization?
- If the problem appears to be related to the optimizer, try the options `-fno-builtin`, `-fno-delayed-branch`, `-fno-schedule-insns`, `-fno-schedule-insns2`, `-fno-omit-frame-pointer` and `-fno-peephole` (in that order).

Sometimes this process indicates a possible mistake on your part because of a misunderstanding of the information in the documentation. Even if the error is on your part, please report the problem if the the compile or runtime error messages or the documentation provided you with inadequate information to avoid the mistake.

Please use the Software Trouble Reports to report any difficulties that you experience with the documentation. We appreciate hearing about any apparent mistakes you find, any information you find lacking in some way, or any sections you find difficult to understand. We also appreciate hearing about anything you particularly like about the documentation.

9.3 Sending Media

If we cannot reproduce a problem because you did not send the necessary software (program source, include file, makefile, shell script, core file, or other crucial file), it will delay answering the STR. Sometimes, we have to close the STR with the answer "not reproducible" or insufficient information." To avoid this, please do the following:

- Include the smallest possible application that demonstrates the problem. This can be a shortened version of the original application. Make sure you send any necessary makefiles shell scripts or other source files needed to reproduce the problem. If you send example applications, make sure that they are runnable programs and not listing files.
- In the case of a program crash, a core dump file may be produced in the directory from which the program was started. If such a file is found, include a copy of it with the STR.
- Include a text file on the medium describing the application sent, a statement of the problem, instructions on how to reproduce the problem, and what you've done to track down the problem. You can send hard copy, but a text file is preferable.
- Clearly label the medium, giving format, contents, density, buffer size, and date.
- Please put all files associated with the problem on tape file 0 in **cpio(4)** format, using a command like this:

```
/bin/ls filenames | cpio -ov > /dev/rmt/0
```

- Use relative instead of absolute pathnames; e.g., program.c instead of /usr/brown/program.c.
- Please use six-inch cartridge tape; density is not selectable. Please place a label on the tape clearly indicating its contents.
- Verify the media and instructions. Make an empty directory in some convenient place (**/usr/tmp**), change to that directory, read the tape in, and follow your instructions:

```
mkdir /usr/tmp/my-str  
cd /usr/tmp/my-str  
cpio -idv < /dev/rmt/0  
# follow your instructions  
rm -rf /usr/tmp/my-str
```

End of GNU C Release Notice

