

**Release Notice:
GNU C Compiler for AViiON
Systems**

Release 1.96.4

Part number 085-600128-04

This release notice applies to the following models:

P001
Q001

Copyright © Data General Corporation, 1989-1992
Unpublished—all rights reserved under the copyright laws of the United States
Printed in the United States of America
February 1992
Licensed material—property of Data General Corporation

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, duplications, 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 [FAR] 52.227-7013 (May 1987).

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 1.96.4

085-600128-04

February 1992

Contents

Introduction	1
Product Description	1
Environment	1
Enhancements and Changes	2
Enhancements	2
Changes	3
Notes and Warnings	3
Notes	3
Documentation	4
Software Distribution	5
Media	5
Organization	5
Files	5
Installation Instructions	5
Preparing a Software Trouble Report (STR)	5
Gathering STR Information	5
Gathering Materials for a Software Problem	6
Sending Media	7

Introduction

This Release Notice describes the GNU C that runs under DG/UX 4.30 or later. In addition, this notice provides information not currently available in the GNU documentation (e.g. warnings and recent enhancements).

This printed release notice always accompanies the software. You may print additional copies of this release notice after you have installed the product. A copy suitable for lineprinters can be found in the file `/usr/release/gcc_1.96.4.rn`.

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.

We provide two means of invoking the GNU C compiler: `/bin/gcc` and `/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.

DG's Release 1.96.4 of GNU C is based on Version 1.96 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 gnuilib or any other libraries supplied with DG/UX on AViiON systems.**

Environment

GNU C Release 1.96.4 runs on an AViiON system running DG/UX Revision 4.30 or later. The GNU C release occupies approximately 3 megabytes of disk space.

Enhancements and Changes

Enhancements

1. On AViiON systems running DG/UX Revision 5.4 or later, **/bin/gcc** is a program that invokes **/usr/lib/gcc/gcc**. (Previously, **/bin/gcc** was a symbolic link to **/usr/lib/gcc/gcc**.) The **/bin/gcc** program supports multiple versions of the GNU C compiler. This release of GNU C is "Version 2" and may be explicitly accessed using the **gcc** option **-V2** and the **cc** option **-KV2**. The command **default-gcc** is also supplied with DG/UX Revision 5.4 or later to query or change the default version on your system.

The **/bin/gcc** program supports the software development environments by supplying the appropriate target predefine (e.g. **-D_DGUXCOFF_TARGET**) for the current environment and by supplying an option that indicates whether the environment is COFF- or ELF-based.

2. Version 1.96.4 of GNU C is the first release of a new implementation of GNU C, currently in limited release from the Free Software Foundation. The new implementation has a number of enhancements, notably in the area of code quality.

Since this version is under active development at the Free Software Foundation, features and options may change or may be dropped in future versions of GNU C.

3. In an ELF environment, the options **-fpic** and **-fPIC** direct GNU C to generate position-independent code for use in shared objects. Use the option **-fPIC** when the linker reports an error indicating that the global offset table has overflowed.
4. The options **-static**, **-shared**, and **-symbolic** are supported in an ELF environment.
5. These options have been added:

- Werror**
- Waggregate-return**
- Wuninitialized**
- Wstrict-prototypes**
- Wmissing-prototypes**
- Wredundant-decls**
- Wnested-externs**
- Wno-parentheses**
- mstandard**
- save-temps**
- fPIC**

6. These options are no longer supported:

- fpcc-struct-return**
- fshared-data**
- fvolatile-global**
- i file**

7. Calls to **strcpy**, **strlen**, **strcmp**, **memcpy**, **memcmp**, **abs**, **fabs**, **labs**, and **alloca** are changed to builtin functions.

Changes

1. The options **~~-mocs-frame-position~~** and **~~-mno-ocs-frame-position~~** are expected to be eliminated in the next version of GNU C.
2. The option **~~-mbig-pic~~** is equivalent to **~~-fPIC~~** and will be eliminated.
3. The options

~~-fsigned-bit~~
~~-fno-signed-bit~~
~~-funsigned-bit~~
~~-fno-unsigned-bit~~

have been replaced with the options

~~-fsigned-bitfields~~
~~-fno-signed-bitfields~~
~~-funsigned-bitfields~~
~~-fno-unsigned-bitfields~~

respectively.

Notes and Warnings

Notes

1. The compiler requires the use of support functions supplied in the library **`/usr/lib/gcc/gcc-gnulib`** for performing block moves as described below and for some operations on the **long long** data type. A conforming OCS application built with GNU C will need to supply a copy of this library.
2. The compiler generates optimal code to perform block moves resulting from structure assignment and some calls to the **memcpy** and **strcpy** functions. Block moves are performed in a forward-going manner, which means that block moves of overlapping regions where the source address is greater than the destination address yield the expected result.

One of three methods is used to perform the block move. For small moves, a straight line sequence of ld/st instructions is generated. For larger moves and moves where the source and destination are known to be properly aligned, a call to one of the internal functions is made. In all other cases, a call to **memcpy** is made. The internal functions begin with the prefix **`__movstr`** and have been added to **`/usr/lib/gcc/gcc-gnulib`** and **`/lib/libc.a`**.

3. Contrary to the OCS, GNU C invoked via **`/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.
4. 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
-D*name*[=*value*]
-E
-H
-I*directory*
-O
-S
-U*name*~~

5. 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**).
6. When you invoke GNU C through **/bin/gcc** without the **-traditional** switch, the compiler will report any inconsistencies, if they exist, in calls to standard library routines. This is because the standard include files include prototype information.
7. Because the m88100 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**).
8. Programs with extremely long initializers take considerable time and swap space to compile.

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 **info(1)** 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 man page **gcc(1)**.

Software Distribution

GNU C accompanies DG/UX.

Media

See the Media Notice for the Client/Server User's Package and/or Operating System User's Package.

Organization

For tape layout, see the Media Notice for the Client/Server User's Package and/or the Operating System User's Package.

Files

The file `/usr/release/gcc_1.96.4.fl` lists the files that constitute the GNU C release.

Installation Instructions

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

Preparing a Software Trouble Report (STR)

Gathering STR Information

If you found an error in the GNU C software or the GNU documentation, or if you have suggestions to make about the product, please fill out and return a Data General Software Trouble Report (STR). (If your contract permits, you may 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. Its 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 version number of the compiler. A line such as

gcc version 1.96 (88open OCS/BCS, DG-1.96.4, Jan 17 1992)

reports version 1.96.4. Reporting version 1.96 is not sufficient.

- Revision of the operating system
 - Names and revisions of other software this product uses
 - The CPU type
 - The 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 was one) of the message. Also, 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.

Gathering Materials for a Software Problem

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

- If the problem occurred while you attempted to compile your program, you should submit with your STR a tape containing the source file of the program and all `#include` files it used, except for the `#include` files supplied with this product and the standard DG/UX system include files in `"/usr/include"`. It is not necessary to send the entire GNU C release, 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) which reproduce the problem. Also, please indicate what shell ("sh" or "csh") you use to execute the program and any special conditions, such as the program only failing 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 which caused the problem suddenly to 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-delayed-branch**, **-fno-schedule-insns**, **-fno-schedule-insns2**, **-fno-omit-frame-pointer** and **-fno-peekhole** (in that order).

Often this process indicates a possible mistake on your part due to such things as a misunderstanding of the information in the documentation. Please report the problem, even if the error appears to be yours, but the compile or runtime error messages or the documentation provided you with inadequate information to avoid the mistake.

We would like you to use Software Trouble Reports freely to report difficulties you may be experiencing 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.

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

1. 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 they are runnable programs, not listing files.

Preparing a Software Trouble Report (STR)

2. 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.
3. 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.
4. Clearly label the medium, giving format, contents, density, buffer size, and date.
5. 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
```

6. Use relative instead of absolute pathnames; e.g., program.c instead of /usr/brown/program.c.
7. Please use 6 inch cartridge tape; density is not selectable. Please place a label on the tape clearly indicating its contents.
8. 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 Release Notice