

Update Notice

GNU C Compiler for AViiON Systems  
Update 1.37.23

August 1990

Part Number 078-600036-00

Models Q001 and P001

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 [FAR] 52.227-7013 (May 1987).

DATA GENERAL CORPORATION  
4400 Computer Drive  
Westboro, Massachusetts 01580

Unpublished - all rights reserved under the copyright laws of the United States.

DG/UX and AViiON are trademarks of Data General Corporation.

Copyright © Data General Corporation 1989-1990.  
All Rights Reserved.

Licensed Material - Property of Data General Corporation.

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

## CONTENTS

1.	Introduction.....	3
2.	Product Description.....	3
3.	Environment.....	3
4.	Enhancements and Changes.....	4
4.1	Enhancements.....	4
4.2	Changes.....	5
5.	Notes and Warnings.....	7
5.1	Notes.....	7
6.	Documentation.....	8
6.1	Documentation Changes.....	9
7.	Software Distribution.....	9
7.1	Media.....	9
7.2	Organization.....	9
7.3	Files.....	9
8.	Installation Instructions.....	9
9.	Preparing a Software Trouble Report.....	10
9.1	Gathering STR Information.....	10
9.2	Gathering Materials for a Software Problem.....	11
9.3	Sending Media.....	12

## 1. Introduction

This Update 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 update notice always accompanies the software. You may print additional copies of this update notice after you have installed the product. A copy suitable for lineprinters can be found in the file /usr/release/gcc\_1.37.23.un.

## 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.

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 Update 1.37.23 of GNU C is based on Revision 1.37 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.**

## 3. Environment

GNU C Update 1.37.23 runs on an AViiON system running DG/UX Revision 4.30 or later. The GNU C update occupies approximately 3 megabytes of disk space.

## 4. Enhancements and Changes

### 4.1 Enhancements

1. The option **'-fvolatile-global'** has been added. When this option is used, references to non-local data items are considered volatile. This can aid the porting of programs using a fragment such as:

```
extern int flag;
while (!flag);
```

where the global data item 'flag' is modified by a signal handler. Without the volatile declaration, the optimizer will eliminate the loop invariant.

2. The option **'-mwarn-changes'** has been added. When this option is used, warning messages are generated for constructs that may result in the generation of an object module that will not work with an object module produced by a previous version of GNU C.

This option identifies the uses and definitions of functions that return a 4-byte aligned 4-byte structure or union as having changed in version 1.37.5. In versions of GNU C prior to 1.37.5, the value was returned by copying the return value to the memory area pointed to by the address contained in r12 at entry to the function. In subsequent versions, the value is returned in r2.

This option identifies a structure or union containing a zero-width bit field of type char or short as potentially having a different layout in version 1.36.3. In versions of GNU C prior to 1.36.3, these members were regarded as having type int. Note that **all** such constructs are identified; not just those that are different.

3. When optimization is specified, the GNU C compiler will normally attempt to fill the delay slot of branches and calls. This can be turned off by use of the **'-fno-delayed-branch'** switch.
4. When optimization is specified, calls to **'strcpy'**, **'memcpy'**, **'strcmp'**, and **'memcmp'** are changed to builtin functions. If **'-Ustrcpy'**, **'-Umemcpy'**, **'-Ustrcmp'**, or **'-Umemcmp'**, are specified, the corresponding builtin function will not be used, even if optimization is specified.
5. The compiler generates optimal code to perform block moves resulting from structure assignment and some calls to the **'memcpy'** and **'strcpy'** functions when optimizing. 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'`.

## 4.2 Changes

1. In some constructs involving the use of a word-aligned, one word structure or union returned by value, the compiler no longer aborts or produces incorrect code. Problems had shown up in uses of the `inet_addr()` function:

```

        sin.sin_addr = inet_addr (name);
and
        *addr = inet_addr (name);

```

2. In traditional mode, a function that returns a float no longer returns a double precision value. Returning a double was an inadvertent change made in 1.36.8.
3. Braces are now optional for that part of an initializer that applies to a union.
4. The preprocessor attempts to handle DOS format files (lines end with `\r\n`) by treating `\r` as a whitespace character. Note that line continuation (`\\\r\n`) does not behave as expected.
5. An enum type with file level scope, followed by one with function prototype scope, becomes visible again after the prototype.
6. The computation of the length of a wide character string (`L"abc"`) is now correct.
7. Code produced when passing a structure that is the result of another function call is now correct. The nested call and its caller were incorrectly sharing the argument area.
8. Code produced for an optimized conversion from float or double to unsigned is now correct.

9. Code produced for an unoptimized post increment no longer has a redundant ld instruction.
10. A compiler abort indicating "Arg Pointer found in PRINT\_OPERAND" when varargs is used with the options '-O' and '-fcombine-regs' is fixed.
11. A compiler abort during the instruction combination pass is fixed.
12. A compiler abort when eliminating the frame pointer is fixed.
13. SDB information for register arguments allocated in a stack slot is now correct.
14. SDB information for anonymous types is now handled correctly when the type is introduced in the parameter declaration.
15. SDB information for variable sized objects now provides the correct base address. The size information is necessarily misleading and may result in unusual displays when debugging.
16. No longer generate an error when an automatic variable with incomplete type is made complete by a later declaration.
17. No longer generate an incorrect warning that signed bit fields are not valid ANSI C.
18. No longer generate an incorrect warning about assignment between incompatible pointer types when one of the types is doubly qualified (signed and volatile) and is pointer to pointer.
19. No longer generate an incorrect warning about comparison of distinct pointer types when one is a qualified pointer to void.
20. Report an error when dereferencing a pointer to an incomplete type (other than an incomplete array type) as required by ANSI C.
21. A warning message is elicited if a data item is initialized and declared 'extern'.
22. The options '-V', '-mocs-frame-position', and '-mno-ocs-frame-position' are expected to be eliminated in the next revision of GNU C. The information provided by the '-V' option can be obtained with the '-v' option.

## 5. Notes and Warnings

### 5.1 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 above 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. 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.
3. 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
-v

```

4. 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'`).
5. 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.

6. 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**')
7. The '**-llib**' option to GNU C will be given to the linker in the same relative location compared to executables, as compared to the source files specified on the '**/bin/gcc**' command line. This means that if a module references items out of the library, the '**-llib**' option should follow the modules that reference it.

## 6. Documentation

Using GNU CC (Part #069-100317-00) documents Data General's version of GNU C. This manual is abridged from Using and Porting GNU CC by Richard M. Stallman of the Free Software Foundation. Using GNU CC includes all information from the original that will be immediately useful to the programmer on the AViiON platform, while eliminating information that is specific to other platforms or to the internals of the compiler. The unabridged manual, in a form suitable for a lineprinter, is in the file `/usr/lib/gcc/gcc.ls` in this update.

The unabridged manual is also available for use with the GNU Emacs Info document browsing subsystem (GNU Emacs is distributed with AViiON Contributed Software, Model# R006AZN20A). In addition, there is a separate info manual for the GNU preprocessor. The unabridged manual is stored as files beginning with the prefix '`/usr/lib/gcc/gcc.info`'. The preprocessor manual is stored as the file '`/usr/lib/gcc/cpp.info`'. To add these files, move all of the files of the manual(s) you wish to install to the '`/usr/opt/emacs/info`' directory, and then edit the file '`dir`' in that directory to add the new menus.

For example, to add these manuals, copy the files '`gcc.info*`' and '`cpp.info`' from the '`/usr/lib/gcc`' directory to the '`/usr/opt/emacs/info`' directory. Then edit the file '`dir`' to include the following lines:

- \* Gcc: (gcc.info).       The GNU C compiler.
- \* Cpp: (cpp.info).       The GNU C preprocessor.

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)`.



## 6.1 Documentation Changes

Changes and additions to the manual Using GNU CC (Part #069-100317-00) are in the file /usr/lib/gcc/doc-update.ls. As suggested in that file, these changes should be noted or inserted in your manual.

## 7. Software Distribution

GNU C accompanies DG/UX.

### 7.1 Media

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

### 7.2 Organization

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

### 7.3 Files

The file /usr/release/gcc\_1.37.23.fl lists the files that constitute the GNU C update.

## 8. 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.

## 9. Preparing a Software Trouble Report

### 9.1 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

GNU CC1 version 1.37.23 (88k, Aug 15 1990)...

reports version 1.37.23. Reporting version 1.37 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.

## 9.2 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 update, 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 four main modes of the compiler:  
**-traditional**, no option, **-ansi**, and **-ansi -pedantic**?
- + 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 **-fno-delayed-branch**, **-fno-omit-frame-pointer** and **-fno-peephole**.

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.

### 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

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.
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 GNU C Update Notice

