Getting Started with the i960® Processor Development Tools

Order Number: 485544-004

Revision	Revision History	Date
-001	Original issue for CTOOLS960 R4.5 and GNU/960 R2.4.	05/94
-002	Original issue for CTOOLS960 & GNU/960 R4.6, and MON R2.1.	12/94
-003	Revision for CTOOLS (formerly CTOOLS960 & GNU/960) R5.0.	02/96
-004	Revision for CTOOLS R5.1.	01/97

In the United States, Intel literature may be obtained by writing:

Literature Distribution Center
Intel Corporation
P.O. Box 7641
Mt. Prospect, IL 60056-7641

Or you can call the following toll-free number:

1-800-548-4725

In locations outside the United States, contact your local Intel sales office.

Intel Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update nor to keep current the information contained in this document.

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.

Intel software products are copyrighted by and shall remain the property of Intel Corporation. Use, duplication or disclosure is subject to restrictions stated in Intel’s Software License Agreement, or in the case of software delivered to the government, in accordance with the software license agreement as defined in FAR 52.227-7013.

No part of this document may be copied or reproduced in any form or by any means without prior written consent of Intel Corporation.

Intel Corporation retains the right to make changes to these specifications at any time, without notice.

Contact your local sales office or distributor to obtain the latest specifications before placing your order.

* Other brands and names are the property of their respective owners.

printed on
recycled paper

Contents

The CTOOLS Software Development Toolset 1

This chapter introduces you to the CTOOLS software development toolset and this manual. Topics include:

- An overview of the toolset
- How to use the printed and online documentation

The CTOOLS Toolset

CTOOLS is a complete C-language software-development toolset for developing embedded applications to run on i960® processors. It contains a C compiler, the gcc960 and ic960 compiler driver programs, an assembler, runtime libraries, a collection of software-development tools and utilities, and printed and on-line documentation. Figure 1-1 shows the various tools and the flow of information among them. Table 1-1 tells you where to find information on each of the CTOOLS programs.

NOTE. *CTOOLS was called CTOOLS960 and GNU/960 in previous releases. Release 5.1 of CTOOLS is the current upgrade of these products. You can install the gcc960 interface to CTOOLS for compatibility with GNU/960, or the ic960 interface for compatibility with CTOOLS960. You also can install both interfaces, one after the other.*

Figure 1-1 CTOOLS Overview

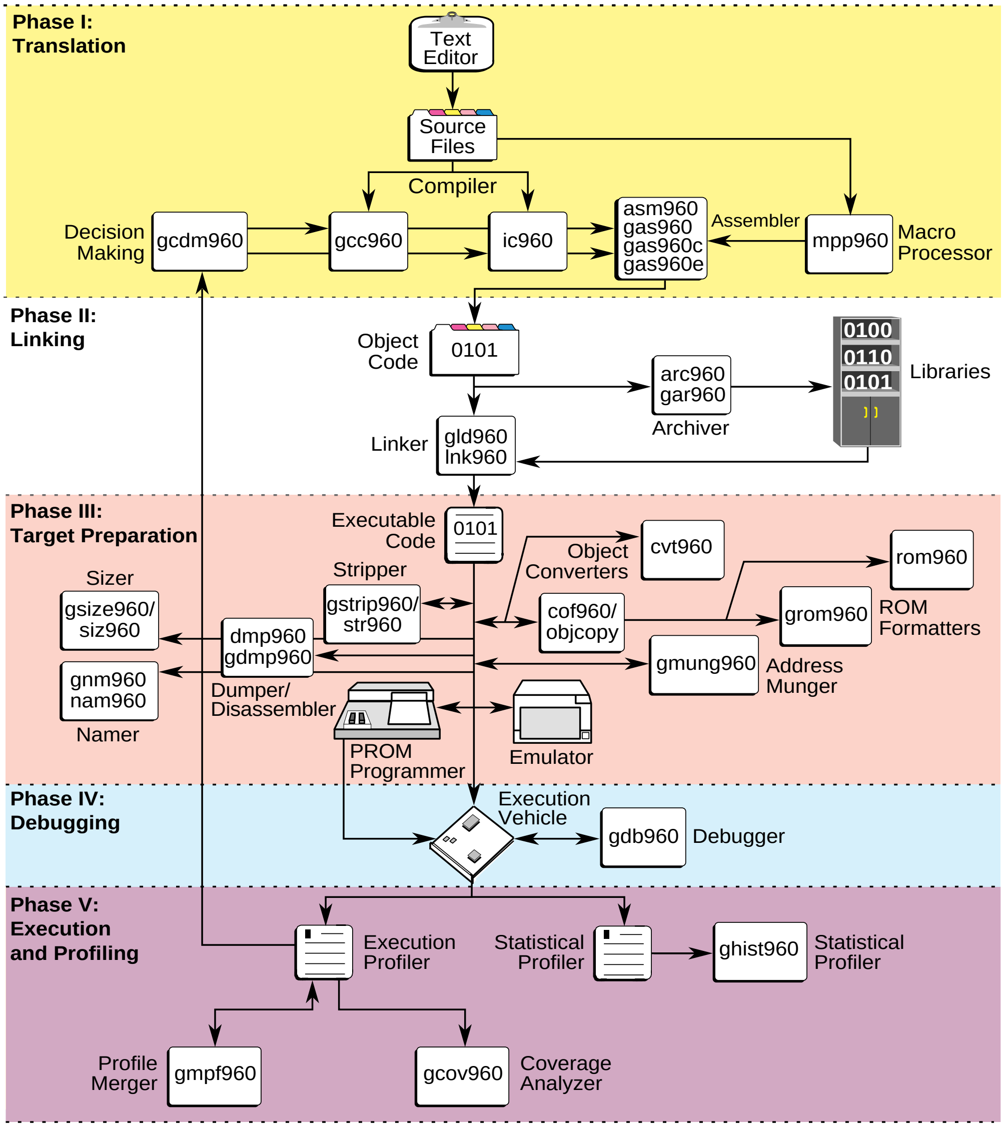

Table 1-1 Where to Find Information on CTOOLS Software

Program Function	Name	Title	Number
archiver	arc960, gar960	*i960 Processor Software Utilities User's Guide*	485277
assembler	asm960, gas960	*i960 Processor Assembler User's Guide*	485276
compiler	gcc960, ic960	*i960 Processor Compiler User's Guide*	651230
converters	cof960/objcopy, cvt960	*i960 Processor Software Utilities User's Guide*	485277
coverage analyzer	gcov960	*i960 Processor Software Utilities User's Guide*	485277
debugger	gdb960	*gdb960 User's Manual*	485546
dumper/ disassembler	dmp960, gdmp960	*i960 Processor Software Utilities User's Guide*	485277
Installer	install, winstall	*Getting Started with the i960 Processor Development Tools* (this manual)	485544
libraries	libc, libm, libh, libfp, libq/libqf, libll, libmon, libhis	*i960 Processor Library Supplement*	651231
linker	gld960, lnk960	*i960 Processor Software Utilities User's Guide*	485277
macro processor	mpp960	*i960 Processor Software Utilities User's Guide*	485277
munger	gmung960	*i960 Processor Software Utilities User's Guide*	485277
name lister	gnm960, nam960	*i960 Processor Software Utilities User's Guide*	485277
ROM image builders	rom960, grom960	*i960 Processor Software Utilities User's Guide*	485277
section size printer	size960, gsize960	*i960 Processor Software Utilities User's Guide*	485277
statistical profiler	ghist960	*i960 Processor Software Utilities User's Guide*	485277
stripper	gstrip960, str960	*i960 Processor Software Utilities User's Guide*	485277

What’s New In CTOOLS

Release 5.1 of the development tools modifies the support for the i960 RP processor, and improves the program development process in several ways:

- Code generation for the i960 RP processor has been changed for performance and future compatibility reasons. New libraries have been added specifically for the i960 RP processor.
- A new graphical user interface to the gdb960 debugger allows source-level debugging with point-and-click ease.
- Windows* NT* is officially supported. The same copies of the tools run under Windows NT 3.51/4.0 and Windows 95. Though the Windows versions of the tools continue to run as command prompt programs, operation under plain MS-DOS is no longer supported.

About the Documentation

The CTOOLS documentation is designed for use with either the gcc960 interface or the ic960 interface, on any supported host. You receive the same set of manuals, the same release notes, and the same online documentation, irrespective of your order code.

Tables 1-2 lists the titles, order numbers and brief descriptions of the CTOOLS manuals. For information on ordering these and other Intel publications, write to:

Intel Literature Sales Department
P.O. Box 7641
Mt. Prospect, IL 60056-7641

You can also order manuals by phone by calling 1-800-548-4725. (International customers should contact their local Intel sales office.) Customers can also request the Customer Literature Guide, order number 210620, for complete information about Intel product literature. Complete CTOOLS manual sets are available. Contact your local Intel sales office or distributor (use order code CTOOLSDOC).

Online Hypertext Documentation

In addition to the printed documentation, a HyperText Markup Language (HTML) hypertext containing the command-line information previously delivered in UNIX*-style manpages is automatically installed in a directory named `html`. This hypertext can be displayed and printed with any HTML-compatible browser, such as Microsoft Internet Explorer or Netscape* Navigator*. To use the hypertext, open the file named `start.htm` in the `html/tools` directory. (**Hint:** in your browser, bookmark this location.)

Table 1-2 Manuals

Title	Number	Contents
Getting Started with the i960 Processor Development Tools (this manual)	485544	Describes installation, environment variables, documentation, and customer service.
i960 Processor Compiler User's Guide	651230	Describes the compiler and two-pass optimization tools.
i960 Processor Library Supplement	651231	Describes the supplied libraries and provides other library information (e.g., all library file names).
i960 Processor Assembler User's Guide	485276	Describes the assembler and assembly language.
i960 Processor Software Utilities User's Guide	485277	Describes the archiver, converters, coverage analyzer, dumper/disassembler, ghist960 profiler, linker, macro processor, namer, rommer, stripper.
gdb960 User's Manual	485546	Describes the gdb960 debugger.
i960 Processor Tools License Guide	614851	Describes CTOOLS licensing and copyrights.

Installation

2

This chapter provides detailed steps for installing CTOOLS on your host system.

Table 2-1 lists the host and operating systems for which CTOOLS R5.1 is available and the media type(s) supplied for each order code. For a list of product contents, see the Release Notes supplied with your toolset. For a list of the manuals supplied with your toolset, see Chapter 1. For information about customer support and product service, see Chapter 3.

Table 2-1 Product Codes, Hosts, Operating Systems, and Media

Product Code	Host System	OS Version	Media
CTOOLSW95R5.1	Intel486™, Pentium®, or Pentium Pro processor-based PC	Windows 95 or Windows NT	CD ROM
CTOOLSUNXR5.1	HP9000/700 workstation	HP-UX 9.05	4 mm & 8 mm DAT tapes; QIC-24 tape
	IBM RS/6000 workstation	AIX 3.2	
	Sun/4 workstation	SunOS 4.1.4 or Solaris 2.5	

Host System Requirements

The amount of disk space required to install CTOOLS depends on which product components you install. You can save a significant amount of disk space by not installing source code for the tools and libraries. Refer to Table 2-2 for a listing of disk space requirements.

Table 2-2 Installation Disk Space Requirements

Operating System	With RTL and Binary Source Code	Without Source Code
UNIX	74 megabytes	26 megabytes
Windows 95/NT	N/A	18 megabytes

Installing CTOOLS

The following sections describe installing the tools on host systems.

Preparing for Installation on UNIX

1. Back up any previously installed versions of the i960 processor language tools.
2. Determine the directory in which to install the tools. Make sure that you have write permission on this directory.
   - If you want to install the new tools in an existing `intel960` directory, you must first remove the tools in this directory.
   - If you want to keep a previous version of the tools, either move them to another directory or choose another location in which to install the new tools.
3. Determine your interrupt key sequence (usually Ctrl-C) by using `stty`.
4. Determine the name of your system's tape device.
5. Place the installation tape in the tape device.

6. Decide if you need to install source code for the tools. The installation script prompts you to enter this information. Having source code for the tools allows you to modify and build the tools, and to receive bug fixes in the form of source patches.

Installing on UNIX Hosts

1. Change directory (`cd`) to an empty or temporary directory.
2. Extract the installation script from the tape device, using the tape device name you determined in step 4 (above).

   To install the ic960 interface to CTOOLS, enter:

   ```
   tar xvf tape_device install.ic
   ```

   To install the gcc960 interface to CTOOLS, enter:

   ```
   tar xvf tape_device install.gnu
   ```
3. To save time during tape scanning, watch for the messages:

   ```
   x install.ic, num bytes, num tape/media blocks
   ```

   or

   ```
   x install.gnu, num bytes, num tape/media blocks
   ```

   After you see this message, you may optionally enter your interrupt key sequence, using the sequence you determined in step 3 (above) and continue to the next step.
4. Execute the installation program.

   To install the ic960 interface to CTOOLS, enter:

   ```
   ./install.ic
   ```

   To install the gcc960 interface to CTOOLS, enter:

   ```
   ./install.gnu
   ```
5. When prompted for the installation directory, specify the directory in which to install the tools. Entering a carriage return specifies the default `/usr/local/intel960`. You can also specify a full path name to install the tools in a custom location (e.g., `/usr/projects/p1/ctools`).

6. If the directory you specify has files in it, a warning similar to the following appears:

   ```
   WARNING: Directory dirname is not empty.
   Installation may corrupt existing files.

   Do you wish to proceed (y/n)?
   Default is [n]
   ```

   Entering `y` may overwrite or remove the old files in the directory and may cause an installation failure. If you enter `n`, the installation script terminates.

   The installation script continues, prompting you for the following information:

   — Tape device (default provided)

   — Host and operating system (table provided)

   — Whether or not to install source code for the tools (default and table provided)

   — Whether or not to install source code for the run-time libraries and other components (defaults or tables provided)

   The installation program invokes `tar` to install the tools. If the installation begins, a message similar to the following displays:

   ```
   Invoking tar for ...
   ```

   If `tar` fails, the following message appears:

   ```
   Tar exited with 1; enter 'abort' or 'retry'.
   ```

   Entering `abort` causes the installation program to terminate. Entering `retry` causes the installation program to try `tar` again.

During installation, the system displays several progress indicator messages related to the renaming of files. No action is required in response to these messages. A message notifies you when the installation is complete.

If you want the tools to be owned by `root` with a Group ID of `bin`, have your system administrator make this change now.

Preparing for Installation on Windows 95/NT

1. Determine the directory in which to install the tools. The installation program places the toolset in `C:\INTEL960`, unless you specify otherwise when the installation program prompts you. (If you want to save a previous version of the tools, you must copy them to a different directory, or specify a different installation directory for the new tools.)
2. You can save disk space by not installing the source code for the runtime libraries.
3. Binary versions of the tools and a UNIX compressed (`.tar`) file of the source code are included on the CD-ROM. You need the following development tools to build the binaries of CTOOLS from source code:
   - UNIX system to modify the filenames to an 8.3 character format
   - Microsoft* Visual* C++ compiler 4.02 or later
   - OPUS Make Version 6.06a for NT or later
   - Thompson Toolkit Version 4.1d or later

NOTE. *Note that you can re-run the installation program later if you need components of the tools that you omit during the initial installation.*

Installing on Windows 95/NT Hosts

To install the tools on a Windows host insert the CD ROM into your drive and enter *`drive`*`:/winstall`. Respond to the prompts as appropriate for your installation.

`winstall` modifies your `autoexec.bat` and saves a copy of the previous version under the name `autoexec.bak`. You must re-boot your system for these settings to take effect.

Preparing to Use the Tools

In order to use the tools, you must set certain environment variables and ensure that certain directories are in your path. The variables that you set vary depending on the interface you selected during installation. The tables below describe the variables that must be set for the GNU/960 and CTOOLS960 interfaces.

Table 2-3 GNU/960 Environment Variables

Variable	Purpose
`G960BASE`	specifies base directory for invoking the CTOOLS programs such as `gcc960` and `gld960`. In Windows, this variable is set by winstall during installation.
`G960ARCH`	specifies the target-architecture libraries.
`G960LIB`	specifies an additional directory for the invocation library and directive-file search path.
`G960LLIB`	library and directive-file search path.

Table 2-4 CTOOLS960 Environment Variables

Variable	Purpose
`I960BASE`	specifies base directory for invoking the CTOOLS programs such as `ic960` or `lnk960`. In Windows, this variable is set by winstall during installation.
`I960ARCH`	specifies the target-architecture libraries.
`I960LIB`	specifies additional directories for the invocation library and directive-file search path.
`I960LLIB`	library and directive-file search path.

Setting Environment Variables

In UNIX you must set the G960BASE or I960BASE for the tools to function correctly. These variables are set during the Windows installaiton. The other described in Tables 2-3 and 2-4 are optional and are provided for your convenience. For example, if you don't set the I960ARCH or G960ARCH variable, you must specify the target architecture on the program command lines with the -A*arch* option. To set the environment variables:

- Windows users must use the set command on the command line or in a batch file such as autoexec.bat. For example, to set the default installation directory:

  ```
  set G960BASE = C:\intel\g960
  ```

- UNIX users must use the set or setenv command in script files, or in a bootup file such as .cshrc, .login, or .profile. You must also specify the host-specific directory under the installation directory. For example, if you installed the toolset in /usr/local/intel on an HP9000/700 workstation, you would use the command:

  ```
  setenv G960BASE = /usr/local/intel/hp700.
  ```

Adjusting Your Path

The directory I960BASE/bin (ic960) or G960BASE/bin (gcc960) must be in your path. In Windows, this modification is made by winstall during the installation process. UNIX users must change their path accordingly.

Table 2-5 lists the environment variables that can be used to alter the default behavior of the tools when using the ic960 interface. Table 2-6 does the same for the gcc960 interface.

Table 2-5 ic960 Interface Environment Variables

Name	Purpose
`I960ARCH`	Specifies code generation architecture. Choices are: `CA`, `CF`, `HA`, `HD`, `HT`, `KA`, `KB`, `SA`, `SB`, `JA`, `JD`, `JF`, `RP`. Default is `KB`.
`I960AS`	Specifies an alternate pathname for the assembler. Default is `I960BASE/bin/asm960`. (`asm960.exe` in Windows.)
`I960BASE`	Specifies top-level directory containing the `bin`, `include`, and `lib` subdirectories. `I960BASE` is necessary for every phase of compilation and linking. The compiler driver uses `I960BASE/lib` to invoke the preprocessor and compiler. The driver uses `I960BASE/bin` to invoke the assembler and linker. The preprocessor uses `I960BASE/include` to find include files. The linker uses `I960BASE/lib` to find libraries, startup modules, and configuration files. `I960BASE` sets defaults for other environment variables in this list. Use these other environment variables to override the paths in `I960BASE`.
`I960CC1`	Specifies an alternate pathname for the compiler. The default is `I960BASE/lib/cc1.960`. (`cc1.exe` in Windows.)
`I960CPP`	Specifies an alternate pathname for the C preprocessor. The default is `I960BASE/lib/cpp.960`. (`cpp.exe` in Windows.)
`I960INC`	Specifies the include file directory. The default is `I960BASE/include`.
`I960LD`	Specifies an alternate linker pathname. The default is `I960BASE/bin/lnk960`. (`lnk960.exe` in Windows.)
`I960LIB`, `I960LLIB`	Specify library search path(s). The default is `I960BASE/lib.`
`I960PDB`	Specifies the profiling program database directory.
`I960ERR`	Windows variable that enables you to redirect errors to `stderr` rather than `stdout` (the default). To use this capability, set `I960ERR` to any string, as in: `set I960ERR="Enable stderr"`
`G960TMP`, `TMP`, or `TMPDIR`	Specifies the directory used for temporary work files. Set it to the name of your temporary file directory.

Table 2-6 gcc960 Interface Environment Variables

Name	Purpose
G960AS	Specifies an alternate pathname for the assembler. Default is G960BASE/bin/gas960. (gas960.exe in Windows.)
G960BASE	Specifies top-level directory containing the bin, include, and lib subdirectories. G960BASE is necessary for every phase of compilation and linking. The compiler driver uses G960BASE/lib to invoke the preprocessor and compiler. The driver uses G960BASE/bin to invoke the assembler and linker. The preprocessor uses G960BASE/include to find include files. The linker uses G960BASE/lib to find libraries, startup modules, and configuration files. G960BASE sets defaults for other environment variables in this list. Use these other environment variables to override the paths in G960BASE.
G960BIN	Specifies an alternate pathname for binary files, such as the assembler and linker. If set, G960BIN overrides G960BASE/bin.
G960CC1	Specifies an alternate pathname for the compiler. The default is G960BASE/lib/cc1.960. (cc1.exe in Windows.)
G960CPP	Specifies an alternate pathname for the C preprocessor. The default is G960BASE/lib/cpp.960. (cpp.exe in Windows.)
G960CRT	Specifies a C runtime startup file as the default.
G960INC	Specifies the include file directory. The default is G960BASE/include.
G960LD	Specifies an alternate linker pathname. The default is G960BASE/bin/gld960. . (gld960.exe in Windows.)
G960LIB	Specify library search path(s). The default is G960BASE/lib.
G960PDB	Specifies the program database directory.
I960ERR	Windows variable that enables you to redirect errors to stderr rather than stdout (the default). To use this capability, set I960ERR to any string, as in: set I960ERR="Enable stderr"
G960TMP, TMP, or TMPDIR	Specifies the directory used for temporary work files. Set it to the name of your temporary file directory.

Customer Support Services 3

Customer Support Services

The Intel Customer Support organization provides the following services to i960 processor development tools customers:

- two types of support contracts
- a World-Wide Web site
- an anonymous FTP server
- a modem-accessible file transfer system

These services are described in the following sections. Following the description of these support services, this chapter also provides information on reporting bugs.

Support Contracts

There are two types of support contracts available for North American customers: Software Assistance and Software Support. Each level of support is available for one-year periods, which can be renewed.

Software Assistance Contracts (DT80960SA)

Software Assistance Contract customers must return the registration card or call 1-800-628-8686 to activate their support. Software Assistance includes:

Hotline assistance	Our telephone hotline, 1-800-628-8686, is open from 7:00 a.m. to 5:00 p.m. Pacific time. Calling the hotline connects you to support engineers who specialize in supporting CTOOLS.

If no support engineer is available when you call, a support engineer will call you within one working hour.

Multiple user support We support up to nine users of your product. You may add users to this support contract by purchasing a *medium* to *large* support contract, listed in Table 3-1.

Free release upgrades Once you register CTOOLS**,** you are automatically updated for all new releases during the 12-month period of your support contract. (Of course, we automatically update you only if you register.)

Software Support Contracts (DT80960SSC)

Software Support Contract customers also receive:

Guaranteed bug fixes/ workarounds North American Software Support customers receive a bug fix or work-around to all critical problems, once they can be duplicated at Intel's development facility. These bugs are placed on a high priority status and receive immediate engineering attention. In some circumstances, Intel offers an engineer at your development site to fix the bug or provide a work-around as quickly as possible.

Inter-release fixes Software Support customers who report a bug receive a fix. (Information on submitting a bug report is included in this manual.) If a bug fix benefits other Software Support customers, we also send each of them the inter-release fix.

A one-year Software Assistance Contract (DT80960SAC) is provided to North American customers with the purchase of CTOOLS. To order additional support services (DT80960SSC) or renew your existing contract, contact your local Intel representative or distributor and ask for order code DT80960SSC (Software Support Contract) or DT80960SAC (Software Assistance Contract). Or call the Support Hotline (1-800-628-8686) to place your order.

The following table lists annual support contract options.

Table 3-1 Annual Support Contracts

Contract	Price	Number of Users Supported
DT80960SAC		
small	$2,500	1-9
medium	$4,500	10-19
large	$6,500	20+
DT80960SSC		
small	$6,000	1-9
medium	$11,000	10-19
large	$15,000	20 +

European Support

European i960 processor tools users are encouraged to report any problems via electronic mail or telephone.

Electronic Mail	`960tools@intel.com`
Toll Free Phone Number	1-800-628-8686
Direct Phone Number[1]	1-916-356-7599

European customers also have these two services available:

BBS:	+44 (0) 793 496340
FaxBACK:	+44 (0) 793 496646

[1] U.S. Phone number. Please use the correct prefix for your country to dial this number.

The Intel World-Wide Web site described in the next section is also available.

Intel's European support is provided on a best effort basis, without contractual obligation to provide a fix or work-around solution.

World-Wide Web Site

The web site provides two areas designed to help CTOOLS customers:

- The i960(R) Processors-Software Support page `http://www.intel.com/design/i960/swsup/`
- The Intel 80960 Software Tools Patches Page `http://www.intel.com/design/i960/patches/`

The software support page provides:

- Software code examples
- Drivers
- Executables
- Other miscellaneous software

The patches page provides:

- Windows 95/NT patched executables
- UNIX patched executables
- UNIX source code
- MON960/MONDB source code

Web site patches provide updated executables and an automated installation process. The Windows 95/NT patch files are self-exploding compressed files. The UNIX binaries include patched executables in compressed tar format. The UNIX source patch files contain updated C code files in UNIX tar format. The source files can be used to rebuild modified tools. UNIX patch files include an automated installation script `ubins`.

Rebuilt or patched tools have a version number of Major.Minor.Fix where:

- Major = 5
- Minor = 1
- Fix = the applied patch release number

For example: 5.1.001.

Note that releases are cumulative, that is, each successive patch release contains all of the previous software updates.

Anonymous FTP

You can retrieve the same software patches described in the previous section via anonymous ftp. To access the ftp system, ftp onto Intel's `ftp.intel.com` machine (the IP address is `134.134.214.7`):

```
$   ftp intel.com
login: anonymous
passwd: use your email address
```

To get the patches for the current release, retrieve the latest readme (`.rm`) file, which describes the software changes. Use the commands:

```
cd /pub/i960/gnu/5.1_patches
binary
get PWyymmdd.rm
```

where “`yymmdd`” represents the name of the readme file associated with the latest patch. If you have any questions, send electronic mail to Customer Support at `960tools@intel.com`.

Bug Fixes

If you report a CTOOLS bug to us, we will attempt to fix it. However, unless you have a support contract, we cannot accept bug reports by telephone. The best way to report bugs is via electronic mail. For a list of addresses, see the How to Report Bugs section that follows.

If you do report a bug to us, please be sure to include a return United States Postal Service address and a phone number, so that we can let you know promptly when we create a fix.

Some of the tools in CTOOLS are based on source code from the Free Software Foundation (FSF). Please note that not all of the tools present in CTOOLS are supported by the FSF, and all of the tools have been extensively modified by Intel, so the FSF may or may not be able to help.

Your bug reports play an important role in making these tools reliable. In order for a bug report to serve its purpose, you must include information we need to fix the bug.

If you are not sure whether you have found a bug, here are some guidelines:

- If any of the tools gets a fatal signal for any valid input, that is a bug.
- If the compiler produces invalid assembly code, for any input whatever (except an `asm` statement), that is a bug, unless the compiler reports errors (not just warnings) that would ordinarily prevent the assembler from being run.
- If the compiler produces valid assembly code that does not correctly execute the input source code, that is a compiler bug.

  However, you must double-check to make sure, because you may have run into an incompatibility between versions of C.

  You may have a program whose behavior is undefined, which happened by chance to give the desired results with another C compiler.

For example, in many nonoptimizing compilers, you can write `x;` at the end of a function instead of `return x;`, with the same results. But the value of the function is undefined if `return` is omitted; it is not a bug when the compiler produces different results.

Problems often result from expressions with two increment operators, as in `f(*p++, *p++)`. Your previous compiler might have interpreted that expression the way you intended; the compiler might interpret it another way. Neither compiler is wrong. The bug is in your code.

After you have localized the error to a single source line, it should be easy to check for these details. If your program is correct and well defined, you have found a compiler bug.

Remember, however, that the definition in C of correct behavior is sometimes quite broad, especially when it concerns program behavior that is not portable between machines, such as:

a. variables that should be and are not marked volatile;

b. default alignment of structure elements;

c. treatment of character variables, signed or unsigned; and

d. undefined behavior surrounding a machine’s subroutine calling sequence.

- If a tool produces an error message for valid input, that is a bug.

  The following is not valid input, and the error message for it is not a bug:

  ```
  int foo (char);

  int
  foo (x)
       char x;
  { ... }
  ```

  The prototype says to pass a `char`, while the definition says to pass an `int` and treat the value as a `char`. This is what the ANSI standard requires.

- If the compiler does not produce an error message for invalid input, that is a compiler bug. However, your idea of "invalid input" might be someone else's idea of "an extension" or "support for traditional practice."

- If you are an experienced user of C compilers, your suggestions for improvement of the compiler are welcome in any case, though they are usually lower priority than fixing actual bugs.

How to Report Bugs

Send bug reports via any of the methods listed in this section. The preferred methods of contact are electronic mail or telephone. Regardless of the method that you use, please include your electronic mail address, phone number, and FAX number with your report.

Via electronic mail:

`960tools@intel.com`

Via FAX:

80960 Development Tool Bug Report: (916) 356-2892

Via telephone:

Customers with either support contract in force may call our support engineers at our HOTLINE at 1-800-628-8686.

The fundamental principle of reporting bugs usefully is this: *report all the facts*. If you are not sure whether to state a fact or leave it out, state it!

To enable us to fix the bug, you should include all these things:

- The version of the tool containing the bug. You can get this by running the tool with the `-v` and `-v960` options. ic960 and gcc960 also place relevant version number information in comments at the beginning of each generated assembler file.

- A complete input file that reproduces the bug. If the bug is in the C preprocessor, send a source file and any header files that it requires. If the bug is in the compiler, run your source file through the C preprocessor by doing:

  ```
  ic960 -P sourcefile.c -o outfile.i
  ```

  or

  ```
  gcc960 -E sourcefile.c > outfile.i
  ```

  and include the contents of `outfile.i` in the bug report. (Any `-I`, `-D` or`-U` options that you used in actual compilation should also be used when doing this.)

  A single statement is not enough of an example. In order to compile it, it must be embedded in a function definition; and the bug might depend on the details of how this is done.
- The command arguments you gave to the tool to observe the bug. To guarantee you won’t omit something important, list them all.
- The type of machine you are using, and the operating system name and version number.
- A description of what behavior you observe that you believe is incorrect. For example, "It gets a fatal signal," or," There is an incorrect assembler instruction in the output."
- If you send examples of assembler output from the compiler, please use `-g` when you make them. The debugging information includes source line numbers that are essential for correlating the output with the input. Embedded comments that highlight the problem are also extremely useful.

If you wish to suggest changes to the compiler source, please send context diffs. If you even discuss something in the compiler source, refer to it by context, not by line number. The line numbers in Intel's development source may not match those in your sources.

License and Warranty A

This appendix indicates where license information is provided, and summarizes the CTOOLS warranty.

License Guide

To make appropriate use of your Intel i960 processor tools, it is important that you know your rights and responsibilities. We urge you to read the *i960 Processor Tools License Guide*, which describes the licensing terms for CTOOLS.

The *i960 Processor Tools License Guide* identifies the various software modules used in the i960 processor tools and provides license information for each software component. The *i960 Processor Tools License Guide* notifies you of the license terms; it is not intended to provide legal advice. Please seek legal counsel or contact Customer Support if you have questions regarding the implications of the license terms.

Warranty and Maintenance

CTOOLS may or may not meet your needs. If, during the first 30 days after you receive these tools, you discover that they do not meet your needs, you may return the entire package, including distribution disks, manuals, and packing materials, and we will refund your distribution fee. Other than this, Intel offers no warranty or guaranteed level of support whatsoever for these products. While we will accept bug reports and attempt to make periodic releases to fix these bugs, unless you purchase specific support services from Intel, we cannot promise to be able to fix any bugs that arise on a particular schedule.

In particular, unless otherwise covered by an applicable Intel Support Agreement,

THERE IS NO WARRANTY FOR ANY PROGRAM, LIBRARIES, OR OTHER SOFTWARE PROVIDED AS PART OF THIS DISTRIBUTION, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS, INTEL CORPORATION, AND/OR OTHER PARTIES PROVIDE THESE PROGRAMS "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THESE PROGRAMS IS WITH YOU. SHOULD THE PROGRAMS PROVE DEFECTIVE, UNDER NO CIRCUMSTANCES SHALL INTEL OR ANY OTHER PARTY BE RESPONSIBLE FOR THE COST OF NECESSARY SERVICING, REPAIR, OR CORRECTION. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL THE COPYRIGHT HOLDERS, INTEL CORPORATION, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR RE-DISTRIBUTE THESE PROGRAMS BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THESE PROGRAMS (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAMS TO OPERATE WITH OTHER PROGRAMS), EVEN IN THE EVENT THAT THE COPYRIGHT HOLDER, INTEL CORPORATION, OR OTHER THIRD PARTIES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Index