B

Installing gawk

This appendix provides instructions for installing *gawk* on Unix-like systems and on PC operating systems.* The primary developer supports GNU/Linux (and Unix), whereas the other ports are contributed. See the section "Reporting Problems and Bugs" later in this chapter for the electronic mail addresses of the people who maintain the respective ports.

The gawk Distribution

This section describes how to get the *gawk* distribution, how to extract it, and then what is in the various files and subdirectories.

Getting the gawk Distribution

There are three ways to get GNU software:

- Copy it from someone else who already has it.
- Order *gawk* directly from the Free Software Foundation. Software distributions are available for Unix, MS-DOS, and VMS, on tape and CD-ROM. Their address is:

  Free Software Foundation
  59 Temple Place, Suite 330
  Boston, MA 02111-1307 USA
  Phone: (617) 542-5942
  Fax (including Japan): (617) 542-2652
  Email: *gnu@gnu.org*
  URL: *http://www.gnu.org*

* See the online Texinfo or Info versions of this book for information about other operating systems. VMS, Amiga, and BeOS have supported ports. Atari and Tandem have unsupported ports.

Ordering from the FSF directly contributes to the support of the foundation and to the production of more free software.

- Retrieve *gawk* by using anonymous *ftp* to the Internet host `gnudist.gnu.org`, in the directory */gnu/gawk*.

The GNU software archive is mirrored around the world. The up-to-date list of mirror sites is available at the main FSF web site *http://www.gnu.org/order/ftp.html*. Try to use one of the mirrors; they will be less busy, and you can usually find one closer to your site.

Extracting the Distribution

gawk is distributed as a `tar` file compressed with the GNU Zip program, `gzip`.

Once you have the distribution (for example, *gawk-3.1.0.tar.gz*), use `gzip` to expand the file and then use `tar` to extract it. You can use the following pipeline to produce the *gawk* distribution:

```
# Under System V, add 'o' to the tar options
gzip -d -c gawk-3.1.0.tar.gz | tar -xvpf -
```

This creates a directory named *gawk-3.1.0* in the current directory.

The distribution filename is of the form *gawk-V.R.P.tar.gz*. The *V* represents the major version of *gawk*, the *R* represents the current release of version *V*, and the *P* represents a *patch level*, meaning that minor bugs have been fixed in the release. The current patch level is 0, but when retrieving distributions, you should get the version with the highest version, release, and patch level. (Note, however, that patch levels greater than or equal to 80 denote "beta" or nonproduction software; you might not want to retrieve such a version unless you don't mind experimenting.) If you are not on a Unix system, you need to make other arrangements for getting and extracting the *gawk* distribution. You should consult a local expert.

Contents of the gawk Distribution

The *gawk* distribution has a number of C source files, documentation files, subdirectories, and files related to the configuration process (see the section "Compiling and Installing gawk on Unix" later in this appendix), as well as several subdirectories related to different non-Unix operating systems:

Various .c, .y, and .h files
The actual *gawk* source code.

README
*README_d/README.**
Descriptive files: *README* for *gawk* under Unix and the rest for the various hardware and software combinations.

INSTALL
A file providing an overview of the configuration and installation process.

ChangeLog
A detailed list of source code changes as bugs are fixed or improvements made.

NEWS
A list of changes to *gawk* since the last release or patch.

COPYING
The GNU General Public License.

FUTURES
A brief list of features and changes being contemplated for future releases, with some indication of the time frame for the feature, based on its difficulty.

LIMITATIONS
A list of those factors that limit *gawk*'s performance. Most of these depend on the hardware or operating system software and are not limits in *gawk* itself.

POSIX.STD
A description of one area in which the POSIX standard for *awk* is incorrect as well as how *gawk* handles the problem.

doc/awkforai.txt
A short article describing why *gawk* is a good language for AI (Artificial Intelligence) programming.

doc/README.card, *doc/ad.block*, *doc/awkcard.in*, *doc/cardfonts*, *doc/colors*, *doc/macros*, *doc/no.colors*, *doc/setter.outline*
The *troff* source for a five-color *awk* reference card. A modern version of *troff* such as GNU *troff* (*groff*) is needed to produce the color version. See the file *README.card* for instructions if you have an older *troff*.

doc/gawk.1
The *troff* source for a manual page describing *gawk*. This is distributed for the convenience of Unix users.

doc/gawk.texi
: The Texinfo source file for this book. It should be processed with TEX to produce a printed document, and with *makeinfo* to produce an Info or HTML file.

doc/gawk.info
: The generated Info file for this book.

doc/gawkinet.texi
: The Texinfo source file for *TCP/IP Internetworking with gawk*. It should be processed with TEX to produce a printed document and with *makeinfo* to produce an Info or HTML file. (This document has been condensed into Chapter 14, *Internetworking with gawk*, for the O'Reilly & Associates edition.)

doc/gawkinet.info
: The generated Info file for *TCP/IP Internetworking with gawk*.

doc/igawk.1
: The *troff* source for a manual page describing the *igawk* program presented in the section "An Easy Way to Use Library Functions" in Chapter 13, *Practical awk Programs*.

doc/Makefile.in
: The input file used during the configuration process to generate the actual *Makefile* for creating the documentation.

Makefile.am
**/Makefile.am*
: Files used by the GNU *automake* software for generating the *Makefile.in* files used by *autoconf* and *configure*.

Makefile.in, *acconfig.h*, *acinclude.m4*, *aclocal.m4*, *configh.in*, *configure.in*, *configure*, *custom.h*, *missing_d/**, *m4/**
: These files and subdirectories are used when configuring *gawk* for various Unix systems. They are explained in the section "Compiling and Installing gawk on Unix" later in this chapter.

*intl/**
*po/**
: The *intl* directory provides the GNU `gettext` library, which implements *gawk*'s internationalization features, while the *po* library contains message translations.

awklib/extract.awk, *awklib/Makefile.am*, *awklib/Makefile.in*, *awklib/eg/**
: The *awklib* directory contains a copy of *extract.awk* (see the section "Extracting Programs from Texinfo Source Files" in Chapter 13), which can be used to extract the sample programs from the Texinfo source file for this book. It also contains a *Makefile.in* file, which *configure* uses to generate a *Makefile*. *Makefile.am* is used by GNU Automake to create *Makefile.in*. The library functions

from Chapter 12, *A Library of awk Functions*, and the *igawk* program from the section "An Easy Way to Use Library Functions" in Chapter 13, are included as ready-to-use files in the *gawk* distribution. They are installed as part of the installation process. The rest of the programs in this book are available in appropriate subdirectories of *awklib/eg*.

*unsupported/atari/**
: Files needed for building *gawk* on an Atari ST (see the online *gawk.info* and *gawk.texi* files in the *gawk* distribution for details).

*unsupported/tandem/**
: Files needed for building *gawk* on a Tandem (see the online *gawk.info* and *gawk.texi* files in the *gawk* distribution for details).

*posix/**
: Files needed for building *gawk* on POSIX-compliant systems.

*pc/**
: Files needed for building *gawk* under MS-DOS, MS Windows and OS/2 (see the section "Installation on PC Operating Systems" later in this appendix, for details).

*vms/**
: Files needed for building *gawk* under VMS (see the online *gawk.info* and *gawk.texi* files in the *gawk* distribution for details).

*test/**
: A test suite for *gawk*. You can use `make check` from the top-level *gawk* directory to run your version of *gawk* against the test suite. If *gawk* successfully passes `make check`, then you can be confident of a successful port.

Compiling and Installing gawk on Unix

Usually, you can compile and install *gawk* by typing only two commands. However, if you use an unusual system, you may need to configure *gawk* for your system yourself.

Compiling gawk for Unix

After you have extracted the *gawk* distribution, *cd* to *gawk-3.1.0*. Like most GNU software, *gawk* is configured automatically for your Unix system by running the *configure* program. This program is a Bourne shell script that is generated automatically using GNU *autoconf*. (The *autoconf* software is described fully in *Autoconf—Generating Automatic Configuration Scripts*, which is available from the Free Software Foundation.)

To configure *gawk*, simply run *configure*:

```
sh ./configure
```

This produces a *Makefile* and *config.h* tailored to your system. The *config.h* file describes various facts about your system. You might want to edit the *Makefile* to change the `CFLAGS` variable, which controls the command-line options that are passed to the C compiler (such as optimization levels or compiling for debugging).

Alternatively, you can add your own values for most *make* variables on the command line, such as `CC` and `CFLAGS`, when running *configure*:

```
CC=cc CFLAGS=-g sh ./configure
```

See the file *INSTALL* in the *gawk* distribution for all the details.

After you have run *configure* and possibly edited the *Makefile*, type:

```
make
```

Shortly thereafter, you should have an executable version of *gawk*. That's all there is to it! To verify that *gawk* is working properly, run `make check`. All of the tests should succeed. If these steps do not work, or if any of the tests fail, check the files in the *README_d* directory to see if you've found a known problem. If the failure is not described there, please send in a bug report (see the section "Reporting Problems and Bugs" later in this appendix.)

Additional Configuration Options

There are several additional options you may use on the *configure* command line when compiling *gawk* from scratch, including:

`--enable-portals`
: Treat pathnames that begin with */p* as BSD portal files when doing two-way I/O with the `|&` operator (see the section "Using gawk with BSD Portals" in Chapter 10, *Advanced Features of gawk*).

`--with-included-gettext`
: Use the version of the `gettext` library that comes with *gawk*. This option should be used on systems that do *not* use Version 2 (or later) of the GNU C library. All known modern GNU/Linux systems use Glibc 2. Use this option on any other system.

`--disable-nls`
: Disable all message-translation facilities. This is usually not desirable, but it may bring you some slight performance improvement. You should also use this option if *––with–included–gettext* doesn't work on your system.

The Configuration Process

This section is of interest only if you know something about using the C language and the Unix operating system.

The source code for *gawk* generally attempts to adhere to formal standards wherever possible. This means that *gawk* uses library routines that are specified by the ISO C standard and by the POSIX operating system interface standard. When using an ISO C compiler, function prototypes are used to help improve the compile-time checking.

Many Unix systems do not support all of either the ISO or the POSIX standards. The *missing_d* subdirectory in the *gawk* distribution contains replacement versions of those functions that are most likely to be missing.

The *config.h* file that *configure* creates contains definitions that describe features of the particular operating system where you are attempting to compile *gawk*. The three things described by this file are: what header files are available, so that they can be correctly included, what (supposedly) standard functions are actually available in your C libraries, and various miscellaneous facts about your variant of Unix. For example, there may not be an `st_blksize` element in the `stat` structure. In this case, `HAVE_ST_BLKSIZE` is undefined.

It is possible for your C compiler to lie to *configure*. It may do so by not exiting with an error when a library function is not available. To get around this, edit the file *custom.h*. Use an `#ifdef` that is appropriate for your system, and either `#define` any constants that *configure* should have defined but didn't, or `#undef` any constants that *configure* defined and should not have. *custom.h* is automatically included by *config.h*.

It is also possible that the *configure* program generated by *autoconf* will not work on your system in some other fashion. If you do have a problem, the file *configure.in* is the input for *autoconf*. You may be able to change this file and generate a new version of *configure* that works on your system (see the section "Reporting Problems and Bugs" later in this appendix, for information on how to report problems in configuring *gawk*). The same mechanism may be used to send in updates to *configure.in* and/or *custom.h*.

Installation on PC Operating Systems

This section covers installation and usage of *gawk* on x86 machines running DOS, any version of Windows, or OS/2. In this section, the term "Win32" refers to any of Windows-95/98/ME/NT/2000.

The limitations of DOS (and DOS shells under Windows or OS/2) has meant that various "DOS extenders" are often used with programs such as *gawk*. The varying

capabilities of Microsoft Windows 3.1 and Win32 can add to the confusion. For an overview of the considerations, please refer to *README_d/README.pc* in the distribution.

Installing a Prepared Distribution for PC Systems

If you have received a binary distribution prepared by the DOS maintainers, then *gawk* and the necessary support files appear under the *gnu* directory, with executables in *gnu/bin*, libraries in *gnu/lib/awk*, and manual pages under *gnu/man*. This is designed for easy installation to a */gnu* directory on your drive—however, the files can be installed anywhere provided AWKPATH is set properly. Regardless of the installation directory, the first line of *igawk.cmd* and *igawk.bat* (in *gnu/bin*) may need to be edited.

The binary distribution contains a separate file describing the contents. In particular, it may include more than one version of the *gawk* executable. OS/2 binary distributions may have a different arrangement, but installation is similar.

Compiling gawk for PC Operating Systems

gawk can be compiled for MS-DOS, Win32, and OS/2 using the GNU development tools from DJ Delorie (DJGPP; MS-DOS only) or Eberhard Mattes (EMX; MS-DOS, Win32 and OS/2). Microsoft Visual C/C++ can be used to build a Win32 version, and Microsoft C/C++ can be used to build 16-bit versions for MS-DOS and OS/2. The file *README_d/README.pc* in the *gawk* distribution contains additional notes, and *pc/Makefile* contains important information on compilation options.

To build *gawk*, copy the files in the *pc* directory (*except* for *ChangeLog*) to the directory with the rest of the *gawk* sources. The *Makefile* contains a configuration section with comments and may need to be edited in order to work with your *make* utility.

The *Makefile* contains a number of targets for building various MS-DOS, Win32, and OS/2 versions. A list of targets is printed if the *make* command is given without a target. As an example, to build *gawk* using the DJGPP tools, enter `make djgpp`.

Using *make* to run the standard tests and to install *gawk* requires additional Unix-like tools, including *sh*, *sed*, and *cp*. In order to run the tests, the *test/*.ok* files may need to be converted so that they have the usual DOS-style end-of-line markers. Most of the tests work properly with Stewartson's shell along with the companion utilities or appropriate GNU utilities. However, some editing of *test/Makefile* is required. It is recommended that you copy the file *pc/Makefile.tst* over the file *test/Makefile* as a replacement. Details can be found in *README_d/README.pc* and in the file *pc/Makefile.tst*.

Using gawk on PC Operating Systems

The OS/2 and MS-DOS versions of *gawk* search for program files as described in the section "The AWKPATH Environment Variable" in Chapter 11, *Running awk and gawk*. However, semicolons (rather than colons) separate elements in the AWKPATH variable. If AWKPATH is not set or is empty, then the default search path is `".;c:/lib/awk;c:/gnu/lib/awk"`.

An *sh*-like shell (as opposed to *command.com* under MS-DOS or *cmd.exe* under OS/2) may be useful for *awk* programming. Ian Stewartson has written an excellent shell for MS-DOS and OS/2, Daisuke Aoyama has ported GNU bash to MS-DOS using the DJGPP tools, and several shells are available for OS/2, including *ksh*. The file *README_d/README.pc* in the *gawk* distribution contains information on these shells. Users of Stewartson's shell on DOS should examine its documentation for handling command lines; in particular, the setting for *gawk* in the shell configuration may need to be changed and the `ignoretype` option may also be of interest.

Under OS/2 and DOS, *gawk* (and many other text programs) silently translate end-of-line `"\r\n"` to `"\n"` on input and `"\n"` to `"\r\n"` on output. A special `BINMODE` variable allows control over these translations and is interpreted as follows:

- If `BINMODE` is `"r"`, or `(BINMODE & 1)` is nonzero, then binary mode is set on read (i.e., no translations on reads).
- If `BINMODE` is `"w"`, or `(BINMODE & 2)` is nonzero, then binary mode is set on write (i.e., no translations on writes).
- If `BINMODE` is `"rw"` or `"wr"`, binary mode is set for both read and write (same as `(BINMODE & 3)`).
- `BINMODE=`*`non-null-string`* is the same as `BINMODE=3` (i.e., no translations on reads or writes). However, *gawk* issues a warning message if the string is not one of `"rw"` or `"wr"`.

The modes for standard input and standard output are set one time only (after the command line is read, but before processing any of the *awk* program). Setting `BINMODE` for standard input or standard output is accomplished by using an appropriate `-v BINMODE=`*`N`* option on the command line. `BINMODE` is set at the time a file or pipe is opened and cannot be changed mid-stream.

The name `BINMODE` was chosen to match *mawk* (see the section "Other Freely Available awk Implementations" later in this appendix). Both *mawk* and *gawk* handle `BINMODE` similarly; however, *mawk* adds a `-W BINMODE=`*`N`* option and an environment variable that can set `BINMODE`, `RS`, and `ORS`. The files *binmode[1-3].awk* (under *gnu/lib/awk* in some of the prepared distributions) have been chosen to match *mawk*'s `-W BINMODE=`*`N`* option. These can be changed or discarded; in

particular, the setting of RS giving the fewest "surprises" is open to debate. *mawk* uses RS = "\r\n" if binary mode is set on read, which is appropriate for files with the DOS-style end-of-line.

To illustrate, the following examples set binary mode on writes for standard output and other files, and set ORS as the "usual" DOS-style end-of-line:

```
gawk -v BINMODE=2 -v ORS="\r\n" ...
```

or:

```
gawk -v BINMODE=w -f binmode2.awk ...
```

These give the same result as the -W BINMODE=2 option in *mawk*. The following changes the record separator to "\r\n" and sets binary mode on reads, but does not affect the mode on standard input:

```
gawk -v RS="\r\n" --source "BEGIN { BINMODE = 1 }" ...
```

or:

```
gawk -f binmode1.awk ...
```

With proper quoting, in the first example the setting of RS can be moved into the BEGIN rule.

Reporting Problems and Bugs

If you have problems with *gawk* or think that you have found a bug, please report it to the developers; we cannot promise to do anything but we might well want to fix it.

Before reporting a bug, make sure you have actually found a real bug. Carefully reread the documentation and see if it really says you can do what you're trying to do. If it's not clear whether you should be able to do something or not, report that too; it's a bug in the documentation!

Before reporting a bug or trying to fix it yourself, try to isolate it to the smallest possible *awk* program and input datafile that reproduces the problem. Then send us the program and datafile, some idea of what kind of Unix system you're using, the compiler you used to compile *gawk*, and the exact results *gawk* gave you. Also say what you expected to occur; this helps us decide whether the problem is really in the documentation.

Once you have a precise problem, send email to *bug-gawk@gnu.org*.

Please include the version number of *gawk* you are using. You can get this information with the command gawk --version. Using this address automatically sends a carbon copy of your mail to me. If necessary, I can be reached directly at

arnold@gnu.org. The bug reporting address is preferred since the email list is archived at the GNU Project. *All email should be in English, since that is my native language.*

Do *not* try to report bugs in *gawk* by posting to the Usenet/Internet newsgroup `comp.lang.awk`. While the *gawk* developers do occasionally read this newsgroup, there is no guarantee that we will see your posting. The steps described above are the official recognized ways for reporting bugs.

Non-bug suggestions are always welcome as well. If you have questions about things that are unclear in the documentation or are just obscure features, ask me; I will try to help you out, although I may not have the time to fix the problem. You can send me electronic mail at the Internet address noted previously.

If you find bugs in one of the non-Unix ports of *gawk*, please send an electronic mail message to the person who maintains that port. They are named in the following list, as well as in the *README* file in the *gawk* distribution. Information in the *README* file should be considered authoritative if it conflicts with this book.

The people maintaining the non-Unix ports of *gawk* are as follows:

Amiga
: Fred Fish, *fnf@ninemoons.com*.

BeOS
: Martin Brown, *mc@whoever.com*.

MS-DOS
: Scott Deifik, *scottd@amgen.com*, and Darrel Hankerson, *hankedr@mail.auburn.edu*.

MS-Windows
: Juan Grigera, *juan@biophnet.unlp.edu.ar*.

OS/2
: Kai Uwe Rommel, *rommel@ars.de*.

Tandem
: Stephen Davies, *scldad@sdc.com.au*.

VMS
: Pat Rankin, *rankin@eql.caltech.edu*.

If your bug is also reproducible under Unix, please send a copy of your report to the *bug-gawk@gnu.org* email list as well.

Other Freely Available awk Implementations

There are three other freely available *awk* implementations. This section briefly describes where to get them:

Unix awk

Brian Kernighan has made his implementation of *awk* freely available. You can retrieve this version via the World Wide Web from his home page.* It is available in several archive formats:

Shell archive
http://cm.bell-labs.com/who/bwk/awk.shar

Compressed tar file
http://cm.bell-labs.com/who/bwk/awk.tar.gz

Zip file
http://cm.bell-labs.com/who/bwk/awk.zip

This version requires an ISO C (1990 standard) compiler; the C compiler from GCC (the GNU Compiler Collection) works quite nicely.

See the section "Extensions in the Bell Laboratories awk" in Appendix A, *The Evolution of the awk Language*, for a list of extensions in this *awk* that are not in POSIX *awk*.

mawk

Michael Brennan has written an independent implementation of *awk*, called *mawk*. It is available under the GPL (see Appendix E, *GNU General Public License*), just as *gawk* is.

You can get it via anonymous *ftp* to the host `ftp.whidbey.net`. Change directory to */pub/brennan*. Use "binary" or "image" mode, and retrieve *mawk1.3.3.tar.gz* (or the latest version that is there).

gunzip may be used to decompress this file. Installation is similar to *gawk*'s (see the section "Compiling and Installing gawk on Unix" earlier in this appendix).

* *http://cm.bell-labs.com/who/bwk/.*

mawk has the following extensions that are not in POSIX *awk*:

- The `fflush` built-in function for flushing buffered output (see the section "Input/Output Functions" in Chapter 8, *Functions*).
- The `**` and `**=` operators (see the section "Arithmetic Operators" and section "Assignment Expressions" in Chapter 5, *Expressions*).
- The use of `func` as an abbreviation for `function` (see the section "Function Definition Syntax" in Chapter 8).
- The `\x` escape sequence (see the section "Escape Sequences" in Chapter 2, *Regular Expressions*).
- The */dev/stdout*, and */dev/stderr* special files (see the section "Special Filenames in gawk" in Chapter 4, *Printing Output*). Use `"-"` instead of `"/dev/stdin"` with *mawk*.
- The ability for `FS` and for the third argument to `split` to be null strings (see the section "Making Each Character a Separate Field" in Chapter 3, *Reading Input Files*).
- The ability to delete all of an array at once with `delete array` (see the section "The delete Statement" in Chapter 7, *Arrays in awk*).
- The ability for `RS` to be a regexp (see the section "How Input Is Split into Records" in Chapter 3).
- The `BINMODE` special variable for non-Unix operating systems (see the section "Using gawk on PC Operating Systems" earlier in this appendix).

The next version of *mawk* will support `nextfile`.

awka

Written by Andrew Sumner, *awka* translates *awk* programs into C, compiles them, and links them with a library of functions that provides the core *awk* functionality. It also has a number of extensions.

The *awk* translator is released under the GPL, and the library is under the LGPL.

To get *awka*, go to its home page at *http://awka.sourceforge.net*. You can reach Andrew Sumner at *andrew_sumner@bigfoot.com*.