

README: Installation instruction for bsplit-2.02

Last update: **Wed Jul 29 08:28:27 1998**

Table of contents

- o [Jump start](#)
- o [Introduction](#)
- o [Installation](#)
- o [UNIX Systems](#)
- o [IBM PC DOS](#)
- o [Microsoft Windows](#)
- o [Test suite](#)
- o [Sample build output for UNIX](#)

Jump start

As with most GNUware, you can build, test, and install this program on most UNIX systems by these simple steps

```
csh et amici:
    setenv CC ...your favorite C or C++ compiler...
    ./configure && make all check install
```

```
sh et amici:
    CC=...your favorite C or C++ compiler...
    export CC
    ./configure && make all check install
```

Or in *one* line, if you have env (most modern UNIX systems do):

```
env CC=... ./configure && make all check install
```

If you don't set the CC environment variable, then gcc (or cc, if gcc is not available) will be assumed.

If you wish to undo a *make install*, just do *make uninstall*; this will remove any files in system directories put there by *make install*.

See below for further details, and for instructions for non-UNIX systems.

Introduction

Please report all problems, suggestions, and comments to the author and maintainer:

Nelson H. F. Beebe

Center for Scientific Computing

University of Utah

Department of Mathematics, 110 LCB

155 S 1400 E RM 233

Salt Lake City, UT 84112-0090

USA

Tel: +1 801 581 5254

FAX: +1 801 581 4148

Email: beebe@math.utah.edu, beebe@acm.org, beebe@computer.org (Internet)

WWW URL: <http://www.math.utah.edu/~beebe/>

Installation

bsplit 2.00 has been updated to use the GNU autoconf automatic configuration system for UNIX installations.

GNU autoconf is run at the maintainer's site to produce the `configure` script from `configure.in`.

The `configure` script is run at each installer's UNIX site to produce `Makefile` from `Makefile.in`, and `config.h` from `config.hin`. The `configure` script is a large (2100+ lines) Bourne shell program that investigates various aspects of the local C implementation, and records its conclusions in `config.h`.

For convenience and safety, the distribution includes a subdirectory named `save` that contains read-only copies of the files `Makefile`, `config.h`, and `configure` created by autoconf and *make configure*. This will allow recovery from a lost or damaged `configure` file.

Should you do a *make maintainer-clean* [**not** recommended, except at the maintainer's site], the `configure` script will be deleted, and you will need recent versions of both GNU m4 and autoconf correctly installed to reconstruct things, which can be done this way:

```
make -f save/Makefile reconfigure
```

Suitable hand-crafted `config.h` files are provided for non-UNIX systems, and in the unlikely event of a failure of the `configure` script on a UNIX system, `config.h` can be manually produced from a copy of `config.hin` with a few seconds of editing work. If you do this, remember to save a copy of your `config.h` under a different name, because running `configure` will destroy it. If you have GNU autoconf installed (the installation is very simple and source code is available from `ftp://prep.ai.mit.edu/pub/gnu/autoconf-x.y.tar.gz`), you might try augmenting `configure.in` instead, then run `autoconf`, `autoheader`, and `configure`.

Thus, on UNIX, installation normally consists of just two steps (assuming a `csh`-compatible shell):

```
setenv CC ...your favorite C or C++ compiler...
./configure && make all check install
```

If you like, add `OPT='your favorite optimization flags'` to the `make` command; by default, no optimization flags are set.

The GNU standard installation directories `/usr/local/bin` for binaries, and `/usr/local/man/man1` for manual pages are assumed. The prefix `/usr/local` can be overridden by providing an alternate definition on the command line:

```
make prefix=/some/other/path install
```

After installation, you can do

```
make distclean
```

to restore the directories to their distribution state. You should also do this between builds for different architectures from the same source tree; *neglecting to do so will almost certainly lead to failure*, because the `config.cache` file created by `configure` will lead to an incorrect `config.h` for the next build.

UNIX Systems

The code can be compiled with either C (K&R or ISO/ANSI Standard C) or C++ compilers. With some C++ compilers, it may be necessary to supply additional switches for force the compiler to stay in C++ mode, rather than reverting to C mode (e.g., on DEC Alpha OSF/1, you must do `setenv CC "cxx -x cxx"`).

On UNIX systems, the only changes that you are likely to need in the `Makefile` are the settings of `CC` and `CFLAGS`, and possibly, `DEFINES`, and if you wish to do *make install*, the settings of `bindir`, `MANDIR`, and `MANEXT`.

If you are installing `bsplit` on a new system, you should definitely run *make check* before installing it on your system. Sample output of *make check* from a UNIX system is given below.

This program has been successfully built and tested with C and C++ compilers on these systems for the 2.00 release (117 builds):

Machine	and model	O/S	Compilers
DEC Alpha	2100-5/250	OSF/1 3.2	/bin/c89, /bin/cc, /bin/cxx -x cxx, /usr/bin/c89, /usr/bin/cc, /usr/ccs/bin/c89, /usr/ccs/bin/cc, /usr/local/bin/g++, /usr/local/bin/gcc, /usr/local/bin/lcc -A -A, /usr/ucb/cc
DECstation	5000/200	ULTRIX 4.3	/bin/cc, /usr/bin/cc, /usr/local/bin/g++, /usr/local/bin/gcc, /usr/local/bin/lcc -A -A
HP	9000/735	HP-UX 10.01	/bin/CC, /bin/c89, /bin/cc, /usr/bin/CC, /usr/bin/c89, /usr/bin/cc, /usr/ccs/bin/cc, /usr/local/bin/g++, /usr/local/bin/gcc
IBM PowerPC	43P	AIX 4.1	/bin/c89, /bin/cc, /bin/xlC, /usr/bin/c89, /usr/bin/cc, /usr/local/bin/g++, /usr/local/bin/gcc
IBM PowerPC	43P	AIX 4.2	/bin/c89, /bin/cc, /bin/xlC, /usr/bin/c89, /usr/bin/cc, /usr/local/bin/g++, /usr/local/bin/gcc
IBM RS/6000	-370	AIX 3.2.5	/bin/c89, /bin/cc, /usr/bin/c89, /usr/bin/cc, /usr/local/bin/gcc
Intel Pentium II	MMX (300MHz)	Linux 2.0.33	/usr/bin/cc, /usr/bin/g++, /usr/bin/gcc
Intel Pentium	MMX (200MHz)	Linux 2.0.30	/usr/bin/cc, /usr/bin/g++, /usr/bin/gcc
NeXT Turbostation	Mach 3.3		/bin/cc, /usr/local/bin/g++, /usr/local/bin/gcc, /usr/local/bin/lcc -A -A
SGI Challenge L	IRIX 5.3		/bin/CC, /bin/cc, /usr/bin/CC, /usr/bin/DCC, /usr/bin/NCC, /usr/bin/cc, /usr/bin/ncc, /usr/local/bin/g++, /usr/local/bin/gcc, /usr/local/bin/lcc -A -A
SGI O2 R10000	-SC	IRIX 6.3	/bin/CC, /bin/c89, /bin/cc, /usr/bin/CC, /usr/bin/DCC, /usr/bin/NCC, /usr/bin/c89, /usr/bin/cc, /usr/local/bin/g++, /usr/local/bin/gcc, /bin/cc -n32, /usr/bin/cc -n32, /bin/DCC -32, /bin/NCC -32, /bin/cc -32, /usr/bin/DCC -32, /usr/bin/NCC -32, /usr/bin/cc -32
SGI Origin/200	-4	IRIX 6.4	/bin/CC, /bin/c89, /bin/cc, /usr/bin/CC, /usr/bin/DCC, /usr/bin/NCC, /usr/bin/c89, /usr/bin/cc, /usr/local/bin/g++, /usr/local/bin/gcc, /bin/cc -n32, /usr/bin/cc -n32, /bin/DCC -32, /bin/NCC -32, /bin/cc -32, /usr/bin/DCC -32, /usr/bin/NCC -32, /usr/bin/cc -32, /bin/cc -64, /usr/bin/cc -64
Sun SPARC	20/512	Solaris 2.6	/opt/SUNWspro/bin/CC, /opt/SUNWspro/bin/c89, /opt/SUNWspro/bin/cc, /usr/local/bin/g++, /usr/local/bin/gcc, /usr/local/bin/lcc -A -A, /usr/ucb/cc
Sun SPARC	4/380	SunOS 4.1.3	/bin/cc, /usr/bin/cc, /usr/lang/CC, /usr/lang/acc, /usr/local/bin/g++, /usr/local/bin/gcc, /usr/local/bin/lcc -A -A,

/usr/ucb/cc

IBM PC DOS

The `ibmpc/dos/README` file contains details of the builds and tests of `bsplit` under several IBM PC DOS C and C++ compilers, and instructions for building and testing `bsplit` with other compilers.

Since `bsplit` uses no floating-point arithmetic, and PC DOS has no shared libraries, I expect that the executables will run on any version of DOS greater than 4.0. They may also run on earlier versions. At the time of writing, MS-DOS 6.22 is current, and the `bsplit` executables work fine on it.

The executable programs for this operating system that are distributed with the 2.02 release are still version 2.00. There are no code changes in 2.02 that affect the operation of the program, and the builds on this system are all done manually, so preparation of a new release requires more human labor than I'm prepared to expend unless absolutely necessary.

Microsoft Windows

With minor changes to version 2.00 to make version 2.01, `bsplit` has been successfully built on Microsoft Windows 98 under both the AT&T U/Win (see <http://www.gtlinc.com/Products/Uwin/Uwin.html> for information; personal, unsupported, non-commercial, licenses of this software are free, subject to acceptance of a license agreement) and Cygnus GNU-Win32 (see <http://www.cygnus.com/misc/gnu-win32>) UNIX-like environments.

Licensing restrictions in U/Win prevent distribution of the executable program in the `ibmpc` subdirectory tree; however, it is trivial to recreate it on a U/Win system with the usual

```
./configure && make all check install
```

provided that a C compiler is available.

Under Cygnus GNU-Win32, where `/bin/sh` does not exist, the build was accomplished by

```
export CONFIG_SHELL=/cygbin/bash.exe
bash < configure && make all check install SHELL=$CONFIG_SHELL
```

The install step failed because the expanded command

```
cp bsplit /usr/local/bin
```

did not copy `bsplit.exe` like it does under U/Win.

These blemishes will no doubt get repaired in newer releases of GNU-Win32.

Test suite

The `bsplit` distribution includes a simple validation test. It is run by

```
make check
```

There should be no output from that test, other than a banner line announcing that expectation.

Please *do* run the validation suite at your site before installing the program. Compilers are complex software systems that also have bugs, so the fact that the program runs correctly somewhere else does not mean that it will do so on a different system.

Sample build output for UNIX

Here is a log of a successful build on Sun Solaris 2.6 using the native C++ compiler, CC:

```
env CC=CC ./configure && make all check install
creating cache ./config.cache
checking whether make sets ${MAKE}... yes
checking for gcc... CC
checking whether the C compiler (CC ) works... yes
checking whether the C compiler (CC ) is a cross-compiler... no
checking whether we are using GNU C... no
checking for working const... no
checking for col... col -x -b
checking for rcp... rcp -p
checking for cp... (cached) rcp -p
checking for cat... cat
checking for chmod... chmod
checking for cmp... cmp
checking for deroff... deroff
checking for distill... distill
checking for dw... dw
checking for geqn... geqn
checking for gzip... gzip
checking for ispell... ispell
checking for ln... ln
checking for ls... ls
checking for man2html... man2html
checking for mkdir... mkdir
checking for mv... mv
checking for groff... groff
checking for rm... rm
checking for sed... sed
checking for sort... sort
checking for spell... spell
checking for gnutar... no
checking for gtar... no
checking for tar... tar
checking for gtbl... gtbl
checking for touch... touch
checking for true... true
checking for unzip... unzip
checking for zip... zip
checking for zoo... zoo
checking for Standard C/C++ function declarations... yes
checking how to run the C preprocessor... CC -E
```

