ANL/MCS-TM-ANL-00/0

Installation Guide for `sowing`, Tools for developing and distributing programs

by

William Gropp and Ewing Lusk

MATHEMATICS AND
COMPUTER SCIENCE
DIVISION

Contents

This *Guide* corresponds to Version 0.1 of `sowing`. It was processed by LaTeX on June 25, 2019.

Abstract

Tool building and distribution is important.

This document describes how to obtain and install `sowing` [1], a collection of portable tools for developing and distributing software. Details on using each of the tools is provided in separate users manuals [2, 3, 4].

1 Quick Start

Here is a set of steps for setting up and minimally testing `sowing`. Details and instructions for a more thorough tour of `sowing`'s features, including installing, validating, benchmarking, and using the tools, are given in the following sections.

1. If you have `gunzip`, get '`sowing.tar.gz`'; otherwise, get '`sowing.tar.Z`' by anonymous `ftp` from `ftp.mcs.anl.gov` in the directory `pub/sowing`.
2. `gunzip -c sowing.tar | tar xovf -` or `zcat sowing.tar.Z | tar xovf -`
3. `cd sowing`
4. `configure` This will attempt to choose an appropriate default architecture and "device" for you. If the defaults are not what you want, see Section 4. If you are *not* going to install sowing, give `configure` the argument `--enable-inplace`.
5. `make >& make.log` (in C-shell syntax). At this point you have built the sowing programs.
6. (Optional) Run the tests (See Section 6 for how to do this).
7. (Optional) If you wish to install `sowing` in a public place so that others may use it, use

   ```
   make install
   ```

   to install `sowing`. Following standard practice with `configure`, the directories for the installation are chosen when you use configure with the `--prefix` and `--datadir` options.

In the following sections we go through these steps in more detail, and describe other aspects of the `sowing` distribution you might want to explore.

2 Obtaining and Unpacking the Distribution

`sowing` can be obtained by anonymous `ftp` from the site `ftp.mcs.anl.gov`. Go to the directory `pub/sowing` and `get` the file `sowing.tar.gz`. This file name is a link to the most recent verstion of `sowing`. Currently it is about 1 Megabyte in size. The file is a `gzipped` tar file, so it may be unpacked with

```
    gunzip -c sowing.tar.gz | tar xvf -
```

If you do not have `gunzip`, but do have `uncompress`, then you must get `sowing.tar.Z` instead, and use either

```
    zcat sowing.tar.Z | tar xvf -
```

or

```
    uncompress sowing.tar.Z
    tar xvf sowing.tar
```

This will create a single directory called `sowing`, containing in various subdirectories the entire distribution, including all of the source code, some documentation (including this *Guide*), `man` pages and the `sowing` environment.In particular, you should see the following files and directories:

`COPYRIGHT` Copyright statement. This code is free but not public domain. It is copyrighted by the University of Chicago.

`Makefile.in` Template for the '`Makefile`', which will be produced when you run `configure`.

`README` Basic information and instructions for configuring.

`aclocal.m4` Used for building '`configure`' from '`configure.in`'; not needed for most installations.

`bin` Home for executable files like `pstogif` and `pstoxbm`.

`configure` The script that you run to create Makefiles throughout the system.

`configure.in` Input to `autoconf` that produces `configure`.

`docs` Documentaiton, including this *Installation Guide* and the users guides.

`include` The include libraries.

`lib` The machine-dependent libraries, after they are built.

`man` Man pages for `MPI`, `MPE`, and internal routines.

`ref` Contains Postscript versions of the `man` pages.

`src` The source code for the portable part of `sowing`. There are subdirectories for the various tools.

If you have problems, check the `sowing` home page on the Web at `http://www.mcs.anl.gov/~gropp/sowing` . This page has pointers to lists of known bugs and patchfiles. If you don't find what you need here, send mail to `mpi-bugs@mcs.anl.gov`.

3 Documentation

This distribution of `sowing` comes with complete `man` pages for the programs. The command `sowingman` in `sowing/bin` is a good interface to the man pages.[1] The `sowing/ref` directory contains printable versions of the man pages for the programs, in compressed PostScript form. The `sowing/docs` directory contains this *Installation Guide* and also the users guides.

4 Configuring sowing

The next step is to configure `sowing` for your particular computing environment. `sowing` can be built for most Unix systems; many of the programs can also be built for Microsoft Windows.

Configuration of `sowing` is done with the `configure` script contained in the top-level directory. This script is automatically generated by the Gnu `autoconf` program from the file `configure.in`, but you do not need to have `autoconf` yourself.

The configure script documents itself in the following way. If you type

```
    configure --help
```

you will get a complete list of arguments and their meanings. So the simplest way to document the options for `configure` is to just show its output here:

```
`configure' configures sowing 1.1.26 to adapt to many kinds of systems.

Usage: ../../configure [OPTION]... [VAR=VALUE]...

To assign environment variables (e.g., CC, CFLAGS...), specify them as
VAR=VALUE.  See below for descriptions of some of the useful variables.

Defaults for the options are specified in brackets.

Configuration:
  -h, --help              display this help and exit
      --help=short        display options specific to this package
      --help=recursive    display the short help of all the included packages
  -V, --version           display version information and exit
  -q, --quiet, --silent   do not print `checking ...' messages
      --cache-file=FILE   cache test results in FILE [disabled]
  -C, --config-cache      alias for `--cache-file=config.cache'
  -n, --no-create         do not create output files
      --srcdir=DIR        find the sources in DIR [configure dir or `..']

Installation directories:
```

[1]The `sowingman` command is created by the configure process described later.

```
  --prefix=PREFIX         install architecture-independent files in PREFIX
                          [/usr/local]
  --exec-prefix=EPREFIX   install architecture-dependent files in EPREFIX
                          [PREFIX]

By default, `make install' will install all the files in
`/usr/local/bin', `/usr/local/lib' etc.  You can specify
an installation prefix other than `/usr/local' using `--prefix',
for instance `--prefix=$HOME'.

For better control, use the options below.

Fine tuning of the installation directories:
  --bindir=DIR            user executables [EPREFIX/bin]
  --sbindir=DIR           system admin executables [EPREFIX/sbin]
  --libexecdir=DIR        program executables [EPREFIX/libexec]
  --sysconfdir=DIR        read-only single-machine data [PREFIX/etc]
  --sharedstatedir=DIR    modifiable architecture-independent data [PREFIX/com]
  --localstatedir=DIR     modifiable single-machine data [PREFIX/var]
  --libdir=DIR            object code libraries [EPREFIX/lib]
  --includedir=DIR        C header files [PREFIX/include]
  --oldincludedir=DIR     C header files for non-gcc [/usr/include]
  --datarootdir=DIR       read-only arch.-independent data root [PREFIX/share]
  --datadir=DIR           read-only architecture-independent data [DATAROOTDIR]
  --infodir=DIR           info documentation [DATAROOTDIR/info]
  --localedir=DIR         locale-dependent data [DATAROOTDIR/locale]
  --mandir=DIR            man documentation [DATAROOTDIR/man]
  --docdir=DIR            documentation root [DATAROOTDIR/doc/sowing-1.1.26]
  --htmldir=DIR           html documentation [DOCDIR]
  --dvidir=DIR            dvi documentation [DOCDIR]
  --pdfdir=DIR            pdf documentation [DOCDIR]
  --psdir=DIR             ps documentation [DOCDIR]

Optional Features:
  --disable-option-checking  ignore unrecognized --enable/--with options
  --disable-FEATURE       do not include FEATURE (same as --enable-FEATURE=no)
  --enable-FEATURE[=ARG]  include FEATURE [ARG=yes]
  --enable-cache          Turn on configure caching
AD_HELP_STRING(--enable-strict, Turn on strict compilation testing when using
GNU-compatible compilers)

--enable-echo   - Turn on strong echoing

--enable-inplace - build the sowing program to run in the
                   distribution directories, not the installation ($ac_default_prefix)
                   directories

--enable-memorycheck - Compile with memory checking code
```

```
Optional Packages:
  --with-PACKAGE[=ARG]    use PACKAGE [ARG=yes]
  --without-PACKAGE       do not use PACKAGE (same as --with-PACKAGE=no)

--with-wwwdir=directory - Specify the root directory for HTML documentation

Some influential environment variables:
  CC          C compiler command
  CFLAGS      C compiler flags
  LDFLAGS     linker flags, e.g. -L<lib dir> if you have libraries in a
              nonstandard directory <lib dir>
  LIBS        libraries to pass to the linker, e.g. -l<library>
  CPPFLAGS    (Objective) C/C++ preprocessor flags, e.g. -I<include dir> if
              you have headers in a nonstandard directory <include dir>
  CXX         C++ compiler command
  CXXFLAGS    C++ compiler flags
  CPP         C preprocessor
  CXXCPP      C++ preprocessor

Use these variables to override the choices made by `configure' or to help
it to find libraries and programs with nonstandard names/locations.

Report bugs to <wgropp@illinois.edu>.
```

Normally, you should use `configure` with as few arguments as you can.

Some sample invocations of `configure` are shown below. In many case, the detailed invocations below are the defaults, which you would get if you invoked `configure` with no arguments.

To build and test in-place, using strict checking with the Gnu C and C++ compilers:

```
configure --enable-strict --datadir=`pwd`/share --prefix=`pwd`
```

5 Compiling sowing

Once `configure` has determined the features of your system, all we have to do now is

```
    make
```

This will clean all the directories of previous object files (if any), compile the source code, build all necessary libraries, and build the programs. If anything goes wrong, check Section 8 to see if there is anything said there about your problem. If not, follow the directions in Section 8.1 for submitting a bug report. To simplify checking for problems, it is a good idea to use

```
    make >& make.log &
```

Specific (non-default) targets can also be made. See the `Makefile` to see what they are.

After running this `make`, the size of the distribution will be about 3 Megabytes (depending on the particular machine it is being compiled for).

6 Thorough Testing

Each program directory contains a `testing` directory contains tests of the program. The command

```
    make testing
```

in the `sowingt` directory will cause these programs to be compiled, linked, executed, and their output to be compared with the expected output. Linking all these test programs takes up considerable space, so you might want to do

```
    make clean
```

afterwards.

If you have a problem, first check the troubleshooting guides and the lists of known problems. If you still need help, send detailed information to `mpi-bugs@mcs.anl.gov`.

7 Installing `sowing` for Others to Use

This step is optional. Once you have tested all parts of the `Sowing` distribution, you may install `sowing` into a publically available directory, and disseminate information for other users, so that everyone can use the shared installation. To install the libraries and include files in a publicly available place, choose a directory, such as `/usr/local/sowing`, change to the top-level `sowing` directory, and do

```
    make install
```

The `man` pages will have been copied with the installation, so you might want to add `/usr/local/mpi/man` to the default system `MANPATH`. The man pages can be conveniently browsed with the `mpiman` command, found in the `sowing/bin` directory.

A good way to handle multiple releases of `sowing` is to install them into directories whose names include the version number and then set a link from `sowing` to that directory. For example, if the current version is 1.0, the installation commands to install into '`/usr/local`' are

```
configure --prefix=/usr/local/sowing-1.0
make
make install
rm /usr/local/sowing
ln -s /usr/local/sowing-1.1.0 /usr/local/sowing
```

7.1 Installing documentation

The SOWING implementation comes with several kinds of documentation. Installers are encouraged to provide site-specific information, such as the location of the installation (particularly if it is not in '`/usr/local/sowing`').

7.1.1 Man pages

A complete set of Unix man pages for the SOWING implementation are in '`sowing/man`'. '`man/man1`' contains the sowing programs.

7.1.2 Web versions of man pages

Web (HTML) versions are available from ftp://ftp.mcs.anl.gov/pub/sowing/sowingwww.tar.Z. They are available at http://www.mcs.anl.gov/sowing/www. A sample Web page is shown below, and is also available in '`docs/install/sowingsite.html`'

```
<HTML>
<TITLE>Documentation for Sowing Tools</TITLE>
<BODY BGCOLOR="FFFFFF">
<H1>Commands</H1>
<MENU>
<LI> <A HREF="www/www1/bfort.html">bfort</A>
<LI> <A HREF="www/www1/doctext.html">doctext</A>
<LI> <A HREF="www/www1/tohtml.html">tohtml</A>
</MENU>
<H1>Manuals</H1>
</BODY>
</HTML>
```

8 Problems

This section describes some commonly encountered problems and their solutions. It also describes machine-dependent considerations. You should also check the Users Guide, where problems related to compiling, linking, and running MPI programs (as opposed to building the SOWING implementation) are discussed.

8.1 Submitting bug reports

Any problem that you can't solve by checking this section should be sent to `mpi-bugs@mcs.anl.gov`. Please include:

- The version of SOWING (e.g., 1.0)
- The output of

```
uname -a
```

for your system. If you are on an SGI system, also

```
hinv
```

- If the problem is with configure run with the `--enable-echo` argument (e.g., `configure --enable-echo`)

If you have more than one problem, please send them in separate messages; this simplifies our handling of problem reports.

The rest of this section contains some information on trouble-shooting `sowing`. Some of these describe problems that are peculiar to some environments and give suggested work-arounds. Each section is organized in question and answer format, with questions that relate to more than one environment (workstation, operating system, etc.) first, followed by questions that are specific to a particular environment. Problems with workstation clusters are collected together as well.

8.2 Problems configuring

8.2.1 General

1. **Q:** The configure reports the compiler as being broken, but there is no problem with the compiler (it runs the test that supposedly failed without trouble).

   A: You may be using the Bash shell (‘`/bin/bash`’) as a replacement for the Bourne shell (‘`/bin/sh`’). We have reports that, at least under LINUX, Bash does not properly handle return codes in expressions. One fix is to use a different shell, such as ‘`/bin/ash`’, on those systems.

   This won't work on some LINUX systems (*every* shell is broken). We have reports that the following will work:

   (a) In ‘`configure`’, change `trap 'rm -f confdefs*' 0` to `trap 'rm -f confdefs*' 1`

   (b) After configure finishes, remove the file ‘`confdefs.h`’ manually.

2. **Q:** configure reports errors of the form

   ```
   checking gnumake... 1: Bad file number
   ```

   A: Some versions of the `bash` shell do not handle output redirection correctly. Either upgrade your version of `bash` or run configure under another shell (such as `/bin/sh`). Make sure that the version of `sh` that you use is not an alias for `bash`. `configure` trys to detect this situation and will normally issue an error message.

8.3 Problems building sowing

8.3.1 General

1. **Q:** When running make on `sowing`, I get this error:

   ```
   ar: write error: No such file or directory
   *** Error code 1
   ```

   I've looked, and all the files are accessible and have the proper permissions.

   A: Check the amount of space in '`/tmp`'. This error is sometimes generated when there is insufficient space in '`/tmp`' to copy the archive (this is a step that `ar` takes when updating a library). The command `df /tmp` will show you how much space is available. Try to insure that at least twice the size of the library is available.

2. **Q:** When running make on `sowing`, I get errors when executing `ranlib`.

   A: Many systems implement `ranlib` with the `ar` command, and they use the '`/tmp`' directory by default because it "seems" obvious that using '`/tmp`' would be faster ('`/tmp`' is often a local disk). Unfortunately, a large number of systems have ridiculously small '`/tmp`' partitions, and making any use of '`/tmp`' is very risky. In some cases, the `ar` commands used by SOWING will succeed because they use the `l` option—this forces `ar` to use the local directory instead of '`/tmp`'. The `ranlib` command, on the other hand, may use '`/tmp`' and cannot be fixed.

   In some cases, you will find that the `ranlib` command is unnecessary. In these cases, you can reconfigure with `-noranlib`. If you must use `ranlib`, either reduce the space used in '`/tmp`' or increase the size of the '`/tmp`' partition (your system administrator will need to do this). There should be at least 20–30 MBytes free in '`/tmp`'.

3. **Q:** When doing the link test, the link fails and does not seem to find any of the sowing system routines:

   ```
     /homes/them/burgess/sowing/lib/IRIX32/ch_p4/mpicc \
                                             -o overtake overtake.o test.o
     ld: WARNING 126: The archive \
        /homes/them/burgess/sowing/lib/IRIX32/ch_p4/libmpi.a \
                                       defines no global symbols. Ignoring.
     ld: WARNING 84: /usr/lib/libsun.a is not used for resolving any symbol.
   ld: ERROR 33: Unresolved data symbol "MPI_COMM_WORLD" -- \
                                             1st referenced by overtake.o.
   ld: ERROR 33: Unresolved text symbol "MPI_Send" -- \
                                             1st referenced by overtake.o.
   ...
   ```

   A: Check that the `ar` and `ranlib` programs are compatible. One site installed the Gnu `ranlib` in such a way that it would be used with the vendors `ar` program, with which it was incompatible. Use the `-noranlib` option to `configure` if this is the case.

8.3.2 SGI

1. **Q:** The build on an SGI Power Challenge fails with

   ```
   Signal: SIGSEGV in Back End Driver phase.
   > ### Error:
   > ### Signal SIGSEGV in phase Back End Driver -- processing aborted
   > f77 ERROR:  /usr/lib64/cmplrs/be died due to signal 4
   > f77 ERROR:  core dumped
   > *** Error code 2 (bu21)
   > *** Error code 1 (bu21)
   > *** Error code 1 (bu21)
   ```

   A: Our information is that setting the environment variable `SGI_CC` to `-ansi` will fix this problem.

8.3.3 DEC ULTRIX

1. **Q:** When trying to build, the `make` aborts early during the cleaning phase:

   ```
   amon:SOWING/sowing>make clean
           /bin/rm -f *.o *~ nupshot
   *** Error code 1
   ```

   A: This is a bug in the shell support on some DEC ULTRIX systems. You may be able to work around this with

   ```
   setenv PROG_ENV SYSTEM_FIVE
   ```

   Configuring with `-make=s5make` may also work.

Acknowledgments

The work described in this report has benefited from conversations with and use by a large number of people. Particular thanks goes to Barry Smith and Lois McInnes for valuable help in the implementation and development of sowing.

References

[1] W. Gropp and E. Lusk. Sowing MPICH: A case study in the dissemination of a portable environment for parallel scientific computing. *The International Journal of Supercomputer Applications and High Performance Computing*, 11(2):103–114, Summer 1997.

[2] William Gropp. Users manual for `bfort`: Producing Fortran interfaces to C source code. Technical Report ANL/MCS-TM-208, Argonne National Laboratory, March 1995.

[3] William Gropp. Users manual for `doctext`: Producing documentation from C source code. Technical Report ANL/MCS-TM-206, Argonne National Laboratory, March 1995.

[4] William Gropp. Users manual for `tohtml`: Producing true hypertext documents from LaTeX. Technical Report ANL/MCS-TM-207, Argonne National Laboratory, March 1995.