Next: Compiling Octave with 64-bit Indexing, Previous: Build Dependencies, Up: Installation   [Contents] [Index]

G.2 Running Configure and Make

- Run the shell script `configure`. This will determine the features your system has (or doesn't have) and create a file named `Makefile` from each of the files named `Makefile.in`.

  Here is a summary of the configure options that are most frequently used when building Octave:

  `--help`

  Print a summary of the options recognized by the configure script.

  `--prefix=`*`prefix`*

  Install Octave in subdirectories below *prefix*. The default value of *prefix* is `/usr/local`.

  `--srcdir=`*`dir`*

  Look for Octave sources in the directory *dir*.

  `--enable-64`

  This is an **experimental** option to enable Octave to use 64-bit integers for array dimensions and indexing on 64-bit platforms. You probably don't want to use this option unless you know what you are doing. See Compiling Octave with 64-bit Indexing, for more details about building Octave with this option.

  `--enable-bounds-check`

  Enable bounds checking for indexing operators in the internal array classes. This option is primarily used for debugging Octave. Building Octave with this option has a negative impact on performance and is not recommended for general use.

  `--disable-docs`

  Disable building all forms of the documentation (Info, PDF, HTML). The default is to build documentation, but your system will need functioning Texinfo and TeX installs for this to succeed.

  `--enable-float-truncate`

  This option allows for truncation of intermediate floating point results in calculations. It is only necessary for certain platforms.

  `--enable-readline`

  Use the readline library to provide for editing of the command line in terminal environments. This option is on by default.

  `--enable-shared`

  Create shared libraries (this is the default). If you are planning to use the dynamic loading features, you will probably want to use this option. It will make your `.oct` files much smaller and on some systems it may be necessary to build shared libraries in order to use dynamically linked functions.

  You may also want to build a shared version of `libstdc++`, if your system doesn't already have one.

  `--enable-dl`

  Use `dlopen` and friends to make Octave capable of dynamically linking externally compiled functions (this is the default if `--enable-shared` is specified). This option only works on systems that actually have these functions. If you plan on using this feature, you should probably also use `--enable-shared` to reduce the size of your `.oct` files.

`--with-blas=<lib>`

By default, configure looks for the best BLAS matrix libraries on your system, including optimized implementations such as the free ATLAS 3.0, as well as vendor-tuned libraries. (The use of an optimized BLAS will generally result in several-times faster matrix operations.) Use this option to specify a particular BLAS library that Octave should use.

`--with-lapack=<lib>`

By default, configure looks for the best LAPACK matrix libraries on your system, including optimized implementations such as the free ATLAS 3.0, as well as vendor-tuned libraries. (The use of an optimized LAPACK will generally result in several-times faster matrix operations.) Use this option to specify a particular LAPACK library that Octave should use.

`--with-magick=<lib>`

Select the library to use for image I/O. The two possible values are `"GraphicsMagick"` (default) or `"ImageMagick"`.

`--with-sepchar=<char>`

Use <char> as the path separation character. This option can help when running Octave on non-Unix systems.

`--without-amd`

Don’t use AMD, disable some sparse matrix functionality.

`--without-camd`

Don’t use CAMD, disable some sparse matrix functionality.

`--without-colamd`

Don’t use COLAMD, disable some sparse matrix functionality.

`--without-ccolamd`

Don’t use CCOLAMD, disable some sparse matrix functionality.

`--without-cholmod`

Don’t use CHOLMOD, disable some sparse matrix functionality.

`--without-curl`

Don’t use the cURL library, disable the ftp objects, `urlread` and `urlwrite` functions.

`--without-cxsparse`

Don’t use CXSPARSE, disable some sparse matrix functionality.

`--without-fftw3`

Use the included FFTPACK library for computing Fast Fourier Transforms instead of the FFTW3 library.

`--without-fftw3f`

Use the included FFTPACK library for computing Fast Fourier Transforms instead of the FFTW3 library when operating on single precision (float) values.

`--without-glpk`

Don’t use the GLPK library for linear programming.

`--without-hdf5`

Don’t use the HDF5 library, disable reading and writing of HDF5 files.

`--without-opengl`

Don’t use OpenGL, disable native graphics toolkit for plotting. You will need `gnuplot` installed in order to make plots.

`--without-qhull`

Don’t use Qhull, disable `delaunay`, `convhull`, and related functions.

`--without-qrupdate`

Don’t use QRUPDATE, disable QR and Cholesky update functions.

`--without-umfpack`

Don’t use UMFPACK, disable some sparse matrix functionality.

`--without-zlib`

Don’t use the zlib library, disable data file compression and support for recent MAT file formats.

`--without-framework-carbon`

Don’t use framework Carbon headers, libraries, or specific source code even if the configure test succeeds (the default is to use Carbon framework if available). This is a platform specific configure option for Mac systems.

`--without-framework-opengl`

Don’t use framework OpenGL headers, libraries, or specific source code even if the configure test succeeds. If this option is given then OpenGL headers and libraries in standard system locations are tested (the default value is `--with-framework-opengl`). This is a platform specific configure option for Mac systems.

See the file `INSTALL` for more general information about the command line options used by configure. That file also contains instructions for compiling in a directory other than the one where the source is located.

- Run make.

  You will need a recent version of GNU Make as Octave relies on certain features not generally available in all versions of make. Modifying Octave’s makefiles to work with other make programs is probably not worth your time; instead, we simply recommend installing GNU Make.

  There are currently two options for plotting in Octave: (1) the external program gnuplot, or (2) the internal graphics engine using OpenGL and FLTK. Gnuplot is a command-driven interactive function plotting program. Gnuplot is copyrighted, but freely distributable. As of Octave release 3.4, gnuplot is the default option for plotting. But, the internal graphics engine is nearly 100% compatible, certainly for most ordinary plots, and users are encouraged to test it. It is anticipated that the internal engine will become the default option at the next major release of Octave.

  To compile Octave, you will need a recent version of `g++` or other ANSI C++ compiler. In addition, you will need a Fortran 77 compiler or `f2c`. If you use `f2c`, you will need a script like `fort77` that works like a normal Fortran compiler by combining `f2c` with your C compiler in a single script.

  If you plan to modify the parser you will also need GNU `bison` and `flex`. If you modify the documentation, you will need GNU Texinfo.

  GNU Make, `gcc` (and `libstdc++`), `gnuplot`, `bison`, `flex`, and Texinfo are all available from many anonymous ftp archives. The primary site is ftp.gnu.org, but it is often very busy. A list of sites that mirror the software on ftp.gnu.org is available by anonymous ftp from ftp://ftp.gnu.org/pub/gnu/GNUinfo/FTP.

Octave requires approximately 1.4 GB of disk storage to unpack and compile from source (significantly less, 400 MB, if you don’t compile with debugging symbols). To compile without debugging symbols try the command

```
make CFLAGS=-O CXXFLAGS=-O LDFLAGS=
```

instead of just `make`.

- If you encounter errors while compiling Octave, first see Installation Problems for a list of known problems and if there is a workaround or solution for your problem. If not, see Trouble for information about how to report bugs.
- Once you have successfully compiled Octave, run `make install`.

  This will install a copy of Octave, its libraries, and its documentation in the destination directory. As distributed, Octave is installed in the following directories. In the table below, *prefix* defaults to `/usr/local`, *version* stands for the current version number of the interpreter, and *arch* is the type of computer on which Octave is installed (for example, ‘`i586-unknown-gnu`’).

  ***prefix*/bin**

  Octave and other binaries that people will want to run directly.

  prefix*/lib/octave-*version

  Libraries like liboctave.a and liboctinterp.a.

  ***prefix*/octave-*version*/include/octave**

  Include files distributed with Octave.

  ***prefix*/share**

  Architecture-independent data files.

  ***prefix*/share/man/man1**

  Unix-style man pages describing Octave.

  ***prefix*/share/info**

  Info files describing Octave.

  ***prefix*/share/octave/*version*/m**

  Function files distributed with Octave. This includes the Octave version, so that multiple versions of Octave may be installed at the same time.

  prefix*/libexec/octave/*version*/exec/*arch

  Executables to be run by Octave rather than the user.

  prefix*/lib/octave/*version*/oct/*arch

  Object files that will be dynamically loaded.

  ***prefix*/share/octave/*version*/imagelib**

  Image files that are distributed with Octave.

Next: Compiling Octave with 64-bit Indexing, Previous: Build Dependencies, Up: Installation [Contents] [Index]