

Next: [Installation Problems](#), Previous: [Running Configure and Make](#), Up: [Installation](#) [[Contents](#)][[Index](#)]

G.3 Compiling Octave with 64-bit Indexing

Note: the following only applies to systems that have 64-bit pointers. Configuring Octave with `--enable-64` cannot magically make a 32-bit system have a 64-bit address space.

On 64-bit systems, Octave is limited to (approximately) the following array sizes when using the default 32-bit indexing mode:

```
double:      16GB
single:      8GB
uint64, int64: 16GB
uint32, int32: 8GB
uint16, int16: 4GB
uint8, int8:  2GB
```

In each case, the limit is really (approximately) 2^{31} elements because of the default type of the value used for indexing arrays (signed 32-bit integer, corresponding to the size of a Fortran INTEGER value).

Trying to create larger arrays will produce the following error:

```
octave:1> a = zeros (1024*1024*1024*3, 1, 'int8');
error: memory exhausted or requested size too large
      for range of Octave's index type --
      trying to return to prompt
```

You will obtain this error even if your system has enough memory to create this array (4 GB in the above case).

To use arrays larger than 2 GB, Octave has to be configured with the option `--enable-64`. This option is experimental and you are encouraged to submit bug reports if you find a problem. With this option, Octave will use 64-bit integers internally for array dimensions and indexing. However, all numerical libraries used by Octave will **also** need to use 64-bit integers for array dimensions and indexing. In most cases, this means they will need to be compiled from source since most (all?) distributions which package these libraries compile them with the default Fortran integer size, which is normally 32-bits wide.

The following instructions were tested with the development version of Octave and GCC 4.3.4 on an x86_64 Debian system.

The versions listed below are the versions used for testing. If newer versions of these packages are available, you should try to use them, although there may be some differences.

All libraries and header files will be installed in subdirectories of `$prefix64` (you must choose the location of this directory).

- BLAS and LAPACK (<http://www.netlib.org/lapack>)

Reference versions for both libraries are included in the reference LAPACK 3.2.1 distribution from [netlib.org](http://www.netlib.org).

- Copy the file `make.inc.example` and name it `make.inc`. The options `-fdefault-integer-8` and `-fPIC` (on 64-bit CPU) have to be added to the variable `OPTS` and `NOOPT`.
- Once you have compiled this library make sure that you use it for compiling Suite Sparse and Octave. In the following we assume that you installed the LAPACK library as `$prefix64/lib/liblapack.a`.

- QRUPDATE (<http://sourceforge.net/projects/qrupdate>)

In the Makeconf file:

- Add `-fdefault-integer-8` to `FFLAGS`.
- Adjust the BLAS and LAPACK variables as needed if your 64-bit aware BLAS and LAPACK libraries are in a non-standard location.
- Set `PREFIX` to the top-level directory of your install tree.

- Run `make solib` to make a shared library.
- Run `make install` to install the library.
- SuiteSparse (<http://www.cise.ufl.edu/research/sparse/SuiteSparse>)

Pass the following options to `make` to enable 64-bit integers for BLAS library calls. On 64-bit Windows systems, use `-DLONGBLAS="long long"` instead.

```
CFLAGS='-DLONGBLAS=long'
CXXFLAGS='-DLONGBLAS=long'
```

The SuiteSparse makefiles don't generate shared libraries. On some systems, you can generate them by doing something as simple as

```
top=$(pwd)
for f in *.a; do
  mkdir tmp
  cd tmp
  ar vx ../$f
  gcc -shared -o ../${f%.a}.so *.o
  cd $top
  rm -rf tmp
done
```

Other systems may require a different solution.

- ATLAS instead of reference BLAS and LAPACK

Suggestions on how to compile ATLAS would be most welcome.

- GLPK
- Qhull (<http://www.qhull.org>)

Both GLPK and Qhull use `int` internally so maximum problem sizes may be limited.

- Octave

Octave's 64-bit index support is activated with the configure option `--enable-64`.

```
./configure \
  LD_LIBRARY_PATH="$prefix64/lib" \
  CPPFLAGS="-I$prefix64/include" LDFLAGS="-L$prefix64/lib" \
  --enable-64
```

You must ensure that all Fortran sources except those in the `liboctave/cruft/ranlib` directory are compiled such that `INTEGERS` are 8-bytes wide. If you are using `gfortran`, the configure script should automatically set the Makefile variable `F77_INTEGER_8_FLAG` to `-fdefault-integer-8`. If you are using another compiler, you must set this variable yourself. You should NOT set this flag in `FFLAGS`, otherwise the files in `liboctave/cruft/ranlib` will be miscompiled.

- Other dependencies

Probably nothing special needs to be done for the following dependencies. If you discover that something does need to be done, please submit a bug report.

- pcre
- zlib
- hdf5
- fftw3
- cURL
- GraphicsMagick++
- OpenGL
- freetype
- fontconfig
- fltk

