Building the OpenGrADS Bundle from Sources

From OpenGrads Wiki

Contents

Which GrADS Version?

The instructions in this page applies to the OpenGrADS Bundle versions 1.10 and 2.0. Although most of the information found here is also valid for the core GrADS

available from COLA (http://grads.iges.org/grads/downloads.html), some of details may differ.

Quick Instructions

Here is the quickest way to build the OpenGrADS Bundle from sources:

- Choose a working directory and make sure you have GNU make.
- Get the supplibs either pre-compiled or compile it yourself. You will need
  - Supplibs v2.2.1 for building GrADS v1.10 or GrADSv2.0.0.oga.x and later
  - Supplibs v2.0.0 for building GrADS v1.9-rc1

Pre-compiled supplibs: Download the binary Supplemental Libraries (Supplibs) from our sf.net site (http://sourceforge.net/project /showfiles.php?group_id=161773&package_id=241681), and untar them in this working directory (replace *2.2.1* below with the actual supplibs version):

```
% gunzip supplibs-2.2.1.tar.gz
% tar xvf supplibs-2.2.1.tar
```

The compiled libraries along with binaries and header files are contained in a platform dependent subdirectory; e.g. `i686-pc-linux-gnu` in case of a Linux build. Provide a symbolic link named `supplibs` to that directory:

```
% ln -s supplibs-2.2.1/i686-pc-linux-gnu supplibs
```

If your platform specific name is different, modify the above line accordingly.
Building the supplibs from sources: Download the *supplibs* sources from our sf.net site (http://sourceforge.net/project /showfiles.php?group_id=161773&package_id=241681), untar them in the same location, and build them:

```
% cd supplibs-2.2.1/src
% gmake install
```

Do not configure or issue a simple make; just do a `gmake install` as indicated above. Once the build has finished successfully, provide a symbolic link named 'supplibs' pointing to the platform dependent directory which contains the newly build libraries:

```
% cd ../..
```

```
% ln -s supplibs-2.2.1/i686-pc-linux-gnu supplibs
```

The name of the installation directory for the supplementary libraries (`i686-pc-linux-gnu` on a typical Linux PC) depends on the build machine's architecture and operating system. You therefore might have to modify the above line accordingly.

- Download the GrADS sources from sf.net (http://sourceforge.net/project/showfiles.php?group_id=161773&package_id=182392), untar them in this same working directory, configure and make:

```
% cd grads-<version>
% ./configure
% gmake
% gmake check
% gmake install     (install binaries)
% gmake binstall    (full OpenGrADS Bundle Install)
```

For checking the extensions:

```
% bundle/bundle_create.sh
% cd extensions
% make
% make check
```

For creating distribution tarballs:

```
% gmake bundle-dist (to create and OpenGrADS Bundle tarball)
% gmake src-dist    (to create a source tarball)
% gmake bin-dist    (binary only tarball)
% gmake gex-dist    (extensions only tarball)
```

Important: To avoid surprises make sure that your supplibs symlink and the GrADS top directory are at the same level:

```
supplibs --> supplibs-2.2.1/i686-pc-linux-gnu
grads-<version>/
```

Read on if you need more information. For installation instructions here.

Introduction

A bundle is a directory in the file system that groups related OpenGrADS resources in one place. Executables, scripts, map and font files are examples of resources that are needed to run GrADS and its utilities. You can find more about

the OpenGrADS Bundle here.

There are three different methods for having GrADS built and installed on your computer:

1. If it is included in your Linux distribution, for example, recent versions of Red Hat and Fedora distributions, install it directly from your CD or from the net with update tools such as `yum`, `apt` or `urpmi`.
2. Install one of the binaries distributions from OpenGRADS download page (http://sourceforge.net/project/showfiles.php?group_id=161773&package_id=182392) at sourceforge. This usually works for a large number of platforms and gives you binaries loaded with all the features. You can also download the basic GrADS binaries (without OpenGrADS extensions) from the GrADS download page (http://grads.iges.org/grads/download) at COLA.
3. If 1) or 2) fail, or for security reasons you prefer to build from sources, this is a viable option as well. The information in this page will guide you through the necessary steps in buiding GrADS from sources: from a fully featured build to a customized build that fits your needs.

Like most applications of this nature, many of GrADS capabilities are implemented by means of external libraries: NetCDF, HDF, OPeNDAP, to name a few. These dependencies are fairly complex packages themselves, and porting them to a new platform require experience with this sort of activity. The good news is, these packages build fairly easily in popular platforms such as Linux, Mac OS X, Unices and even Windows. The GrADS code itself is very compact and easily portable. If you are going to have a problem, it will most likely be building one of the external packages, not GrADS.

There are a few options to satisfy GrADS external dependencies:

1. The GrADS developers maintain a central repository with source code for these packages, with binaries available for the supported platforms. See Supplemental Libraries (Supplibs) for more information.
2. Your OS distributions may provide development version for a number of these libraries. For Linux in particular, you would want to install packages with names having *-devel* in it as these will include the header files necessary to compile GrADS.
3. Or else, you download the sources from the package website and do the build yourself. Compared to using sources from the *supplibs* in 1) the only risk of this method is that you may end up using a version of the package that is not quite compatible with GrADS. You have been advised.

However, except for the X11 client libraries, you do not need to resolve all the external dependences. If the GrADS `configure` script cannot find the necessary package, it will disable the GrADS feature that required that package. The next

section will help identifying which packages are relevant to build GrADS with the features that you need.

Deciding which external libraries you need

GrADS has built in support for reading binary and GRIB-1 files, and support for saving graphics metafiles or encapsulated postscript images. You can also write binary data with the `set gxout fwrite` graphical output option. Of this writing, the X Windows client libraries are required and enables you to do interactive graphics, including animation. Other optional features can be enabled depending on which external libraries you have available. The table below summarizes the relatinship between each external library and the features each enables.

Table 1. GrADS features enabled by external packages.

If you have...	You will be able to...
readline, ncurses/termcap	perform command line editing, an almost essential feature
Xaw, libsx	produce Athena widget graphical user interface (GAGUI)
png, jpeg, gd	produce PNG, JPEG image output with printim
tiff, geotiff	produce geotiff and kml output
grib2c, jasper	read GRIB-2 files
NetCDF-3, udunits	read and write NetCDF-3 files
HDF-4, zlib, udunits	read HDF-4 and NetCDF-3 files, write HDF-4 files
LibDAP, curl, xml2, zlib, NetCDF-4, udunits	read NetCDF-3, and OpenDAP/Gridded data, write NetCDF-3
... + HDF-5	... plus NetCDF-4/HDF-5
... + gadap	... plus OPeNDAP station data

So, decide which features you need and use Table 1 to determine which of the dependencies you need to supply. Generally, binary distributions of the supplibs will fulfill all these requirements.

Getting the sources

The source code for the official releases of GrADS can be obtained from the GrADS download page (http://grads.iges.org/grads/download) at COLA. The OpenGrADS Bundle releases are available from our sourceforge site:

```
http://sourceforge.net/project/showfiles.php?group_id=161773&package_id=182392
```

Doing the build

Using the Supplibs

See the #Quick Instructions above. Starting from pre-compiled supplibs usually make the process of building GrADS quite easy. Just make sure that you untar the supplibs at directory parallel to your sources, and provide a symbolic link `supplibs` pointing to the platform dependent directory within `supplibs-2.1.0` containing the binary supplibs. Alternatively, you may set the environment variable `SUPPLIBS` with the location of your supplibs. For more information, enter

```
% configure --help
```

to obtain a description of other configuration parameters such as compilers and libraries. Usually you don't have to set these parameters, but they may handy if you want to tweak the build somehow. Also note that building GrADS v1.10 and v2.0.a5 requires the use of `supplibs-2.1.0`.

Platform specific notes follow.

FreeBSD 6.3/DesktopBSD 1.6

- Updated `gcc` to 4.2, including `gfortran`
- Set the following environment variables before running `configure`:

```
export CC=gcc42
export CXX=g++42
export XAW7_CFLAGS=/usr/local/include
export XAW7_LIBS=/usr/local/lib/libXaw7.a
```

Then as usual

```
./configure
gmake
gmake check
```

Mac OS X

When setting up the supplibs, symlink the platform specific folder to `supplibs`, e.g.,

```
% ln -s supplibs-2.0.0/i686-apple-darwin8.10.1 supplibs
```

In general, supplibs to not need to rebuilt with every minor OS upgrade. For example, supplibs built for `i686-apple-darwin8.10.1` work just fine for `i686-apple-darwin8.11.1`.

Building the Dependencies from Scratch

My suggestion would be for you to build the supplibs from sources, as it contains all that is necessary and the build mechanism ensures that all the libraries are built cooperatively. A single tarball with sources for the supplibs are available here:

```
http://sourceforge.net/project/showfiles.php?group_id=161773&package_id=241681
```

If you insist in downloading the sources from the developers of each library, no problem. I'd still suggest starting with the versions provided in the supplibs, or else get these tarballs (ftp://grads.iges.org/grads/Supplibs/2.0/src/) from COLA.

Satisfying dependences with packages in your OS distribution

The main point to remember here is that you will need development versions of each library, usually having `-devel` in the package name. A few Linux distrbutions are discussed bellow. Please help us improve this documentation by providing us with notes for additional distributions. When configuring enter:

```
 ./configure --enable-dyn-supplibs
```

Fedora

Many dependencies of grads are available in the fedora repository (since Fedora 3, in fedora extras and in main fedora after the merge, in fedora 7):

- udunits-devel
- readline-devel
- zlib-devel

- libjpeg-devel
- gd-devel
- libXaw-devel, Xaw3d-devel or neXtaw-devel
- libsx-devel
- hdf-devel
- netcdf-devel
- libnc-dap-devel
- libdap-devel

With a monolithic X (in RHEL/Centos 4, fedora up to fedora 4), the X libraries development package is `xorg-x11-devel`. In recent fedora distributions, the X lib packages that may be needed are `libXmu-devel` and `libX11-devel`.

For RHEL/Centos most of these packages are in the EPEL repository or other third party repository (although it is often recommended not to mix incompatible third party repositories).

Another important note is that since fedora 5, `wgrib` is a separate package.

Arch

Arch Linux (http://www.archlinux.org) is a lightweight and flexible linux distribution that tries to Keep It Simple.

As it is a rolling distro, there is no such thing as a release (for easy installation You can have a snapshot four to six times each year).

You can find the GrADS sources in the **AUR** repository at http://aur.archlinux.org/packages.php?do_Details=1&ID=13458. Following the **Arch way** of making packages, You need to:

- Download the tarball at http://aur.archlinux.org/packages/opengrads-cvs/opengrads-cvs.tar.gz
- Extract files

```
tar zxvf opengrads-cvs.tar.gz
```

- Enter into the opengrads-cvs directory and use makepkg

```
cd opengrads-cvs
makepkg
```

- To install **repository** packages needed but not installed, You can use

```
makepkg -s
```

- libnc-dap is in the **AUR** repository as well, so repeat the procedure for the libnc-dap first.
- netcdf-pre is in the **AUR** repository as well, so if You need the gradsnc4 executable, repeat the procedure for the hdf5-pre and netcdf-pre first.

All the other dependencies are in the community or extra repositories.

If everithing goes OK at the end, the package `opengrads-cvs-YYYYMMDD-X.[i686-x86_64].pkg.tar.gz` should be in the directory or in `PKGDEST` as configured in `/etc/makepkg.conf` ready for installation with `pacman`.

```
pacman -U opengrads-cvs*.pkg.tar.gz
```

More in-depth guide on building packages for **Arch** can be found at: http://wiki.archlinux.org/index.php/Makepkg. Helper programs such as `yaourt` (http://www.archlinux.fr/yaourt-en) can help in making **Arch** packages. Please note that also in **Arch** wgrib is a separate package: http://aur.archlinux.org /packages.php?do_Details=1&ID=7696

MacOS X: Fink

Please help improve the documentation by providing us with this information.

MacOS X: MacPorts

Please help improve the documentation by providing us with this information.

Mandriva

Please help improve the documentation by providing us with this information.

Debian

Please help improve the documentation by providing us with this information.

Ubuntu

Please help improve the documentation by providing us with this information.

SuSe

Please help improve the documentation by providing us with this information.

OpenBSD

Please help improve the documentation by providing us with this information.

FreeBSD

Please help improve the documentation by providing us with this information.

NetBSD

Please help improve the documentation by providing us with this information.

Building the OpenGrADS Extensions

Starting with versions 2.0.a5 (and with the legacy version 1.10) the OpenGrADS extensions are built alongside the core GrADS sources under `src/`. While the main GrADS build is based on the GNU autotools, the extensions have their own build mechanism which is not based on the autotools. Instead, it relies on GNU make extensions for automatic detection of compiler flags and library discovery.

At the top `extensions/` directory, the GNUmakefile there recursively build the extension shared libraries in each subdirectory. The are 2 makefile fragments at this level:

gex.mk
: This fragment is not included but *inheritted by each GNUmakefile in the subdirectories. It has code for detection C and Fortran compilers, including system dependent flags for making the extension shared libraries work within GrADS.*

suplibs.mk
: This fragment has code to locate the supplibs, if present.

Building the supplibs does not require GrADS to be built, not even configured. As of this writing, most of the extensions use the so-called Level 0 API, and as such includes `grads.h` and must be used with a version of GrADS corresponding to this `grads.h`. More importantly, the main data structures must be the same. Any mismatch in size or order of the elements in the data structures will cause the extension and/or GrADS to malfunction or even to crash. Future versions of the extensions will be designed not to depend so much on the internals details of GrADS.

To build a particular extension is sufficient to go the particular directory, e.g.,

```
% cd extensions/hello
```

and issue

```
% make
```

To test the particular extension, just start GrADS and load the UDX table,

```
% ../../src/grads -l
...
ga-> load udxt hello.udxt
```

Contact Information

If you would like to help us improving this documentation please drop a note at the opengrads-devel@lists.sourceforge.net (mailto:opengrads-devel@lists.sourceforge.net) list.

- This page was last modified on 28 October 2011, at 07:01.