Obtaining NARR, CFSRR and other grib data from the web,
the NOMADS way

NOMADS is a collection of techniques for obtaining grib data. No one method is the best because people have different constraints (programming, bandwidth, computer resources).

Some examples:

- Mr P wants all variables for all levels for all times for the globe
- Mr. B has very limited bandwidth
- Mr. S wants the 2 meter temperature for Boston on 1/1/2000
  wants something compatible with common programs

Mr. P: is a pig and wants to download terrabytes of data

- allow user to select the needed fields, levels, and times
- use compressed data: grib2
- minimal use of server CPU and I/O bandwidth
- Partial http downloading (aka “Fast grib downloading”)
  - http allows you to read bytes N-M of file (web page)
  - bytes N-M can be the desired grib messages
  - very efficient for the http server
- easy to script

Mr. B: is on a bandwidth budget

- allow user to select the needed fields, levels and times
- regional subsetting (takes CPU resources on server)
- compressed data: grib2
- grib2: g2sub (aka, grib_filter)
  - a URL will return the selected subset of a grib2 file
- easy to script
- g2sub is a front end for wgrib2

Mr. S: wants a small amount of data

- can use “partial http downloads” or g2sub but want something easier to use
- easy to use = compatible with common programs

- CPU and compression is not an issue
- OPeNDAP (using GrADS Data Server, GDS)
    - GDS has to run on the server
    - requests get sent to GDS
    - GDS reads the index file for the file or collection
    - GDS reads selected records from the selected grib file
    - sends data back to the client
    - most overhead
    - based on GrADS
    - standard protocol
    - supported by many clients, even spreadsheets

- alternative is the web plot package
    - not on the NCEP's operational Nomad server
    - point and click
    - front end for GrADS
    - can download time series

Partial HTTP downloading

Partial http downloading is a very efficient method of downloading grib data. You download a text inventory of the grib file, select the desired fields and then do a random-access read of the grib file. All these actions are with standard HTTP reads.

For this technique to work, the web server needs to have a wgrib/wgrib2 inventory on the web server. Typically, they have the .inv or .idx suffix. You obtain the inventory by the command

```
get_inv.pl URL_of_inventory
```

The get_inv.pl program gets the inventory from the server and adds a byte range field.

The next step is filtering the inventory to select the desired fields. This step is identical to the filtering step in wgrib/wgrib2. Finally, the last step is to take this filtered inventory, extract the desired byte ranges and pass it onto the curl command. The basic command looks like

```
get_inv.pl $URL.inv | grep (filter) | get_data.pl $URL output.grb
```

Here is an example of a script that loops over a range of dates and gets some NARR from the NCDC nomads server.

```
#!/bin/sh
#
# sample scripts that downloads selected fields from NCDC's NARR archive
#
# range of dates to download (every 3 hours)
date0=2006010118
date1=2006010203

# check for bad dates
date=$date0
if [ $date -lt 1979010100 ] ; then
  echo "bad date YYYYMMDDHH $date"
  exit 88
fi
if [ $date -gt 2018010100 ] ; then
  echo "bad date YYYYMMDDHH $date"
  exit 88
fi
while [ $date -le $date1 ]
do
#  check if file had already been downloaded
   if [ ! -f narr_${date}.grb ] ; then
#     don't over load the server
      sleep 1
      echo ">> $date"
      yymmdd=`echo $date | cut -c1-8`
      yymm=`echo $date | cut -c1-6`
      hr=`echo $date | cut -c9-10`

      URL0="http://nomads.ncdc.noaa.gov/data/narr/$yymm/$yymmdd"
      URL="$URL0/narr-a_221_${yymmdd}_${hr}00_000"

#     $URL = location of grib file
#     $URL.inv = location of inventory file
#     this code downloads the file
      get_inv.pl "$URL.inv" | \
         egrep ':(TMP|UGRD|VGRD|PRES:sfc):' | \
         egrep -v ':(max wind lev|tropopause|cld top):' | \
         get_grib.pl "$URL.grb"  narr_$yymmdd${hr}.grb >/dev/null
   fi

#   increment by 3 hours
#   this version requires GNU date command
    d="`echo $date | cut -c1-8` `echo $date | cut -c9-10`"
    date=`date --date="$d +3 hours" "+%Y%m%d%H"`
done
```

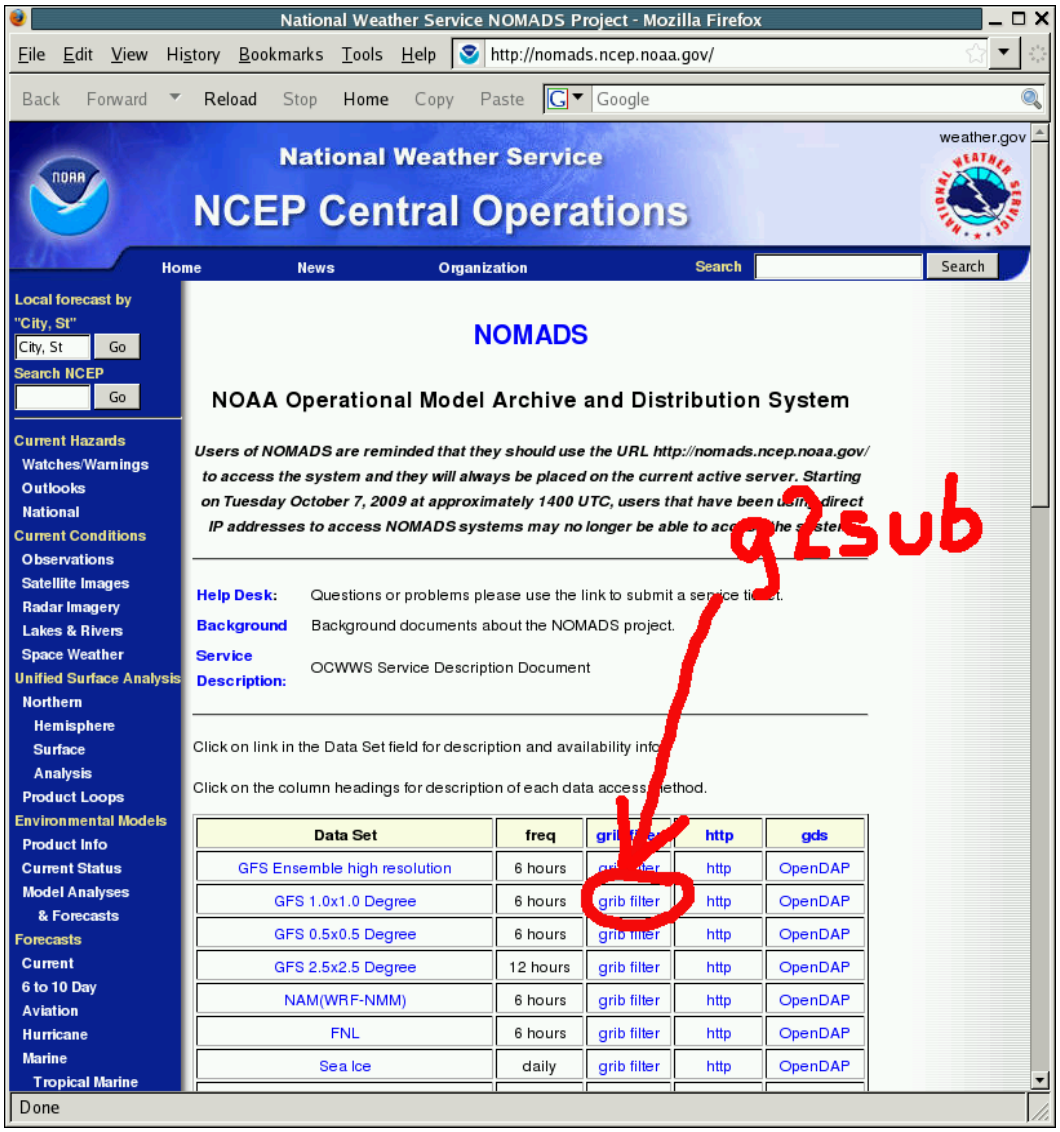
Minimal bandwidth: g2subset/grib-filter

The “partial http downloads” retrievs the entire grid which is “wasteful” if you only want a small regional sub-domain. (For example, downloading the global fields to get the boundary conditions for regional model of New England.) On the nomads servers, you use g2subset (also known as grib-filter).

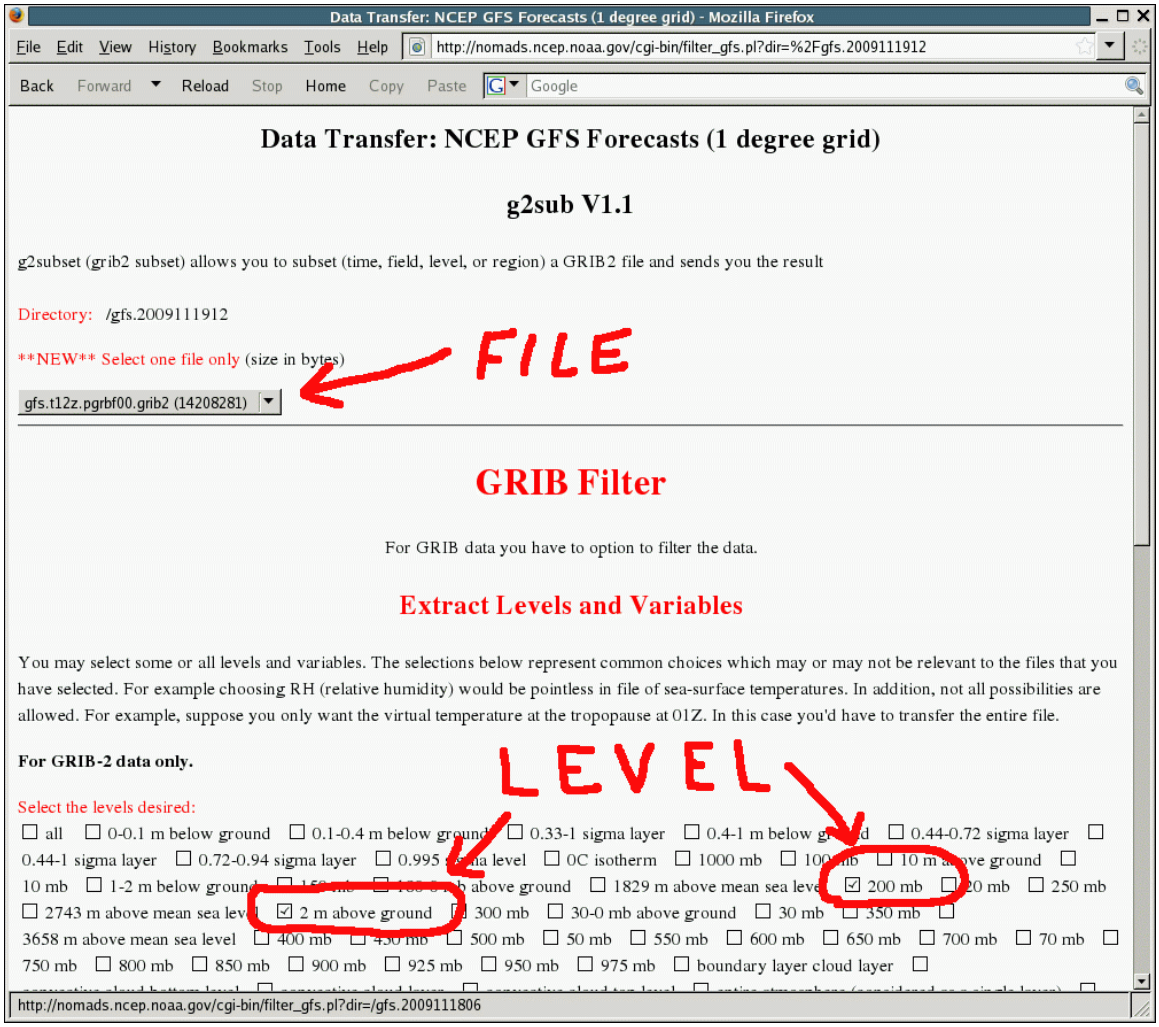
- grib2 files only
- filters for time, level, variable
- can extract latitude-longitude boxes
- grib2 output (system configured compression)
- easy to script

At the time of writing, there isn't a NOMADS site serving CFSRR data. We will use the GFS forecasts for the example. On the operational NCEP NOMADS site

http://nomads.ncep.noaa.gov

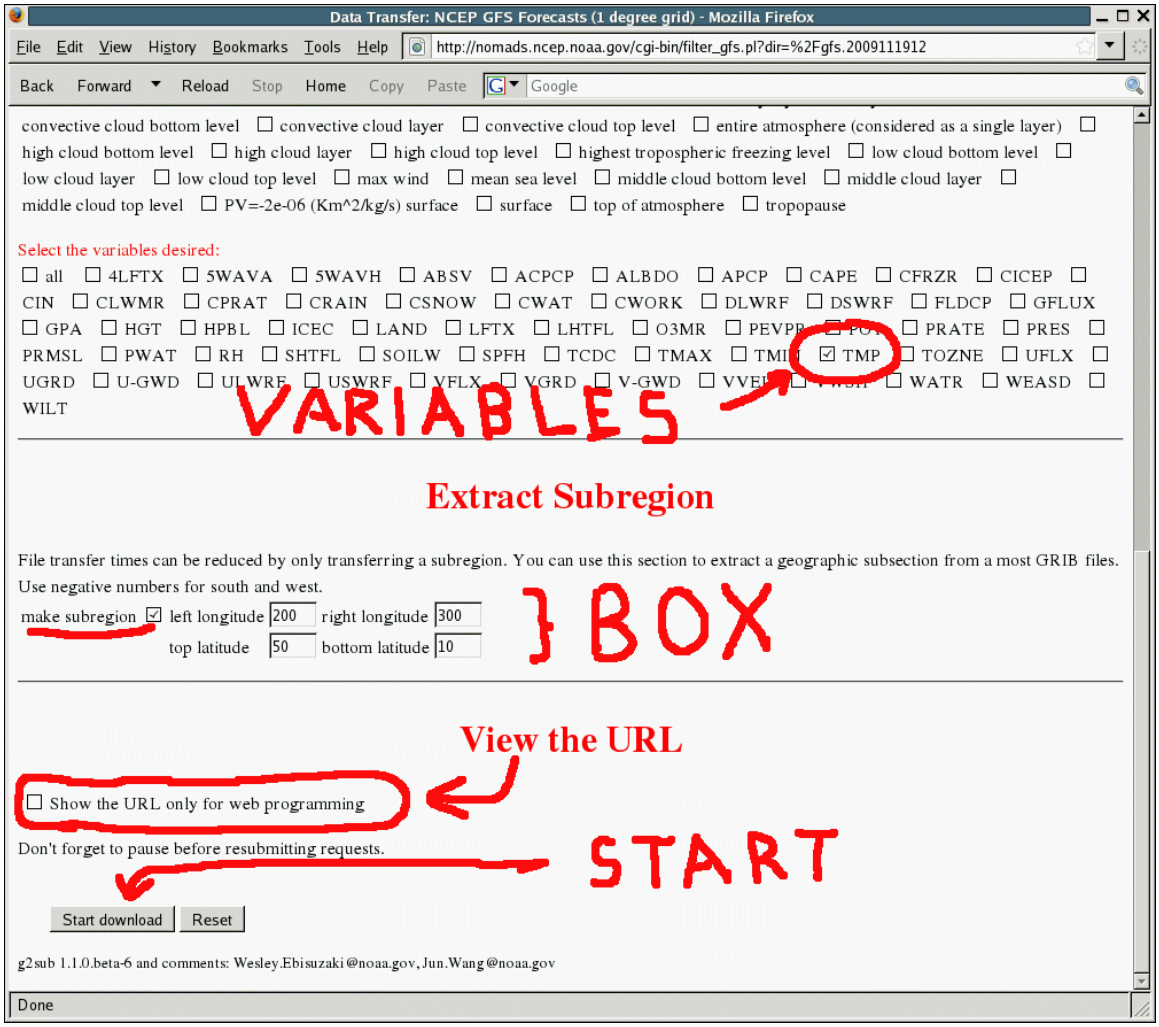


By clicking on “grib-filter”, we'll get the setup page(s) for the download. After selecting the directory, we'll see a page like this,



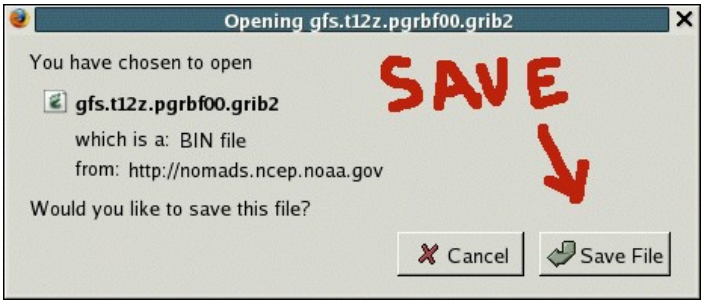
First you select the file, levels, and variables. Note that not all variables are on all levels. For example, if you ask for the soil moisture at 200 mb, you'll get no matches. If you see a file called ocean or "ocn", don't expect to find the 500 mb height. Look for a file with the word "pgb" or "pgrb" which is short for pressure-grib. To save time, you can select more than one variable and level at a time. In our example, the 2 meter and 200 mb TMP have been selected. TMP is the grib name for temperature. You can find the translations of the grib names in,

http://www.nco.ncep.noaa.gov/pmb/docs/grib2/grib2_table4-1.shtml
http://www.nco.ncep.noaa.gov/pmb/docs/grib2/grib2_table4-2-0-0.shtml



To get a subregion, select “make subregion” and enter the parameters for the box. Use negative latitudes for the western and and southern hemispheres. Also make sure the top latitude is numerically greater than the bottom latitude.

For our example, don't click on “Show the URL only for web programming” and click on “Start download” instead. With firefox, a popup will ask you if you want to save your file. Selecting “Save File” will write the grib file to your machine.



Here is the inventory of the downloaded file.

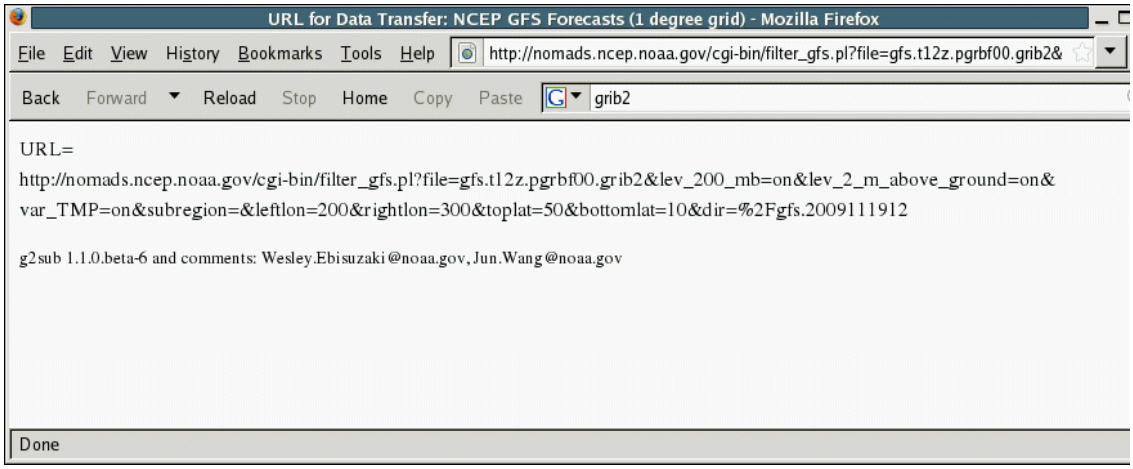
```
-sh-3.00$ wgrib2 -V gfs.t12z.pgrbf00.grib2
1:0:vt=2009111912:200 mb:anl:TMP Temperature [K]:
    ndata=4141:undef=0:mean=217.027:min=207.5:max=229.2
    grid_template=0:
        lat-lon grid:(101 x 41) units 1e-06 input WE:SN output WE:SN res 48
        lat 10.000000 to 50.000000 by 1.000000
        lon 200.000000 to 300.000000 by 1.000000 #points=4141

2:4320:vt=2009111912:2 m above ground:anl:TMP Temperature [K]:
    ndata=4141:undef=0:mean=289.093:min=249.36:max=302.18
    grid_template=0:
        lat-lon grid:(101 x 41) units 1e-06 input WE:SN output WE:SN res 48
        lat 10.000000 to 50.000000 by 1.000000
        lon 200.000000 to 300.000000 by 1.000000 #points=4141
```

As you can see from the inventory, only two fields were downloaded using the small grid.

Automating the Download

After a thousand downloads, you realize there has to be a better way. Then you remember the option “Show the URL only for web programming”. It has to be important because it's circled in red in a previous figure. Selecting “Show the URL” and then click on “Start Download” gives you the page with the URL.



Knowing the URL allows you to download with one line at the shell. You can use cURL or wget.

```
-sh-3.00$ URL="http://nomads.ncep.noaa.gov/cgi-bin/filter_gfs.pl?
                file=gfs.t12z.pgrbf00.grib2&.(text deleted)"
-sh-3.00$ curl “$URL” -o my_grib_file1
-sh-3.00$ wget -O my_grib_file2 “$URL”
```

The structure of the URL is straight forward. The question marks indicates the start of the arguments and the arguments are separated by ampersands.

http://nomads.ncep.noaa.gov/	machine name
/cgi-bin/filter_gfs.pl	cgi-bin program name
file=gfs.t12z.pgrbf00.grib2	name of file to downloaded. "t12z" indicates the forecast started at 12 hours UTC. "f00" indicates the file is a 0 hour forecast.
lev_200_mb=on	get level "200 mb"
lev_2_m_above_ground	get level "2 m above ground"
var_TMP=on	get variable "TMP"
subregion=on	regional subset is "on"
leftlon=200	left longitude is set to 160W
rightlon=300	right longitude is set to 60W
toplat=50	top latitude is set to 50N
bottomlat=10	bottom latitude is set to 10N
dir=%2Fgfs.2009111912	%2F is the hex code for / use subdirectory ./gfs.2009111912

In our example, you can change the starting time of the forecast by altering the "dir=" argument and the hour in the file name (t12z).

Automating a download from a web page is a new task for most users and some new users will have some difficulty. If you are having difficulty, just remember that most of the downloads from the NOMADS sites are automated and people have done it from both windows and linux/unix environments. Our code works on the shell script level but that is not the only way. Some users have embedded the download into their code rather than use curl or wget.

Good Practices with Automated Downloads

Some NOMADS sites have blacklists. Once on the blacklist, your machine will be ignored at the firewall. This is necessary to keep the site functioning.

#1 Offense:

do until downloaded

```
        download file
    enddo
```

Suppose the file doesn't exist, the requesting site can send hundreds of failed requests/second which will waste server resouces. The considerate user will do the following.

```
do until downloaded
        download file
        if (not downloaded) sleep (appropriate time interval)
enddo
```

Offense #2

Trying to download a hundred files at the same time.

The NOMADS web site is a shared resource, don't try to get more than your share. Disks are faster when read sequentially (one file, next file, next file) rather than in random order (part of file1, part of file2, .., part of file 100, part of file 1, etc).

OPeNDAP

Nomads includes GDS (GrADS Data Server) which an OPeNDAP server. From opendap.org,

OPeNDAP: Open-source Project for a Network Data Access Protocol

OPeNDAP is a framework that simplifies all aspects of scientific data networking.

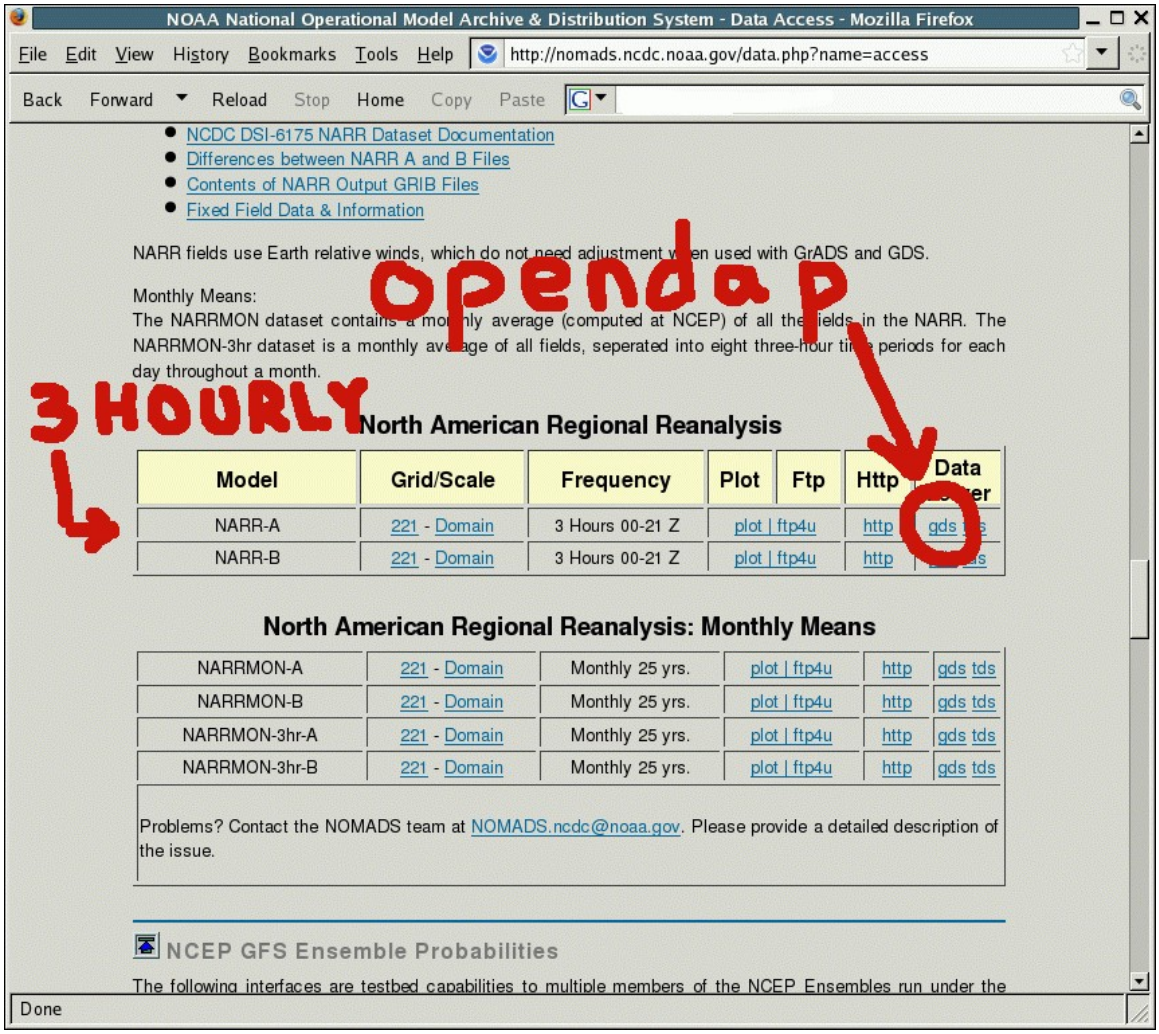
OPeNDAP provides software which makes local data accessible to remote locations regardless of local storage format. OPeNDAP also provides tools for transforming existing applications into OPeNDAP clients (i.e., enabling them to remotely access OPeNDAP served data).

If you have software that is OPeNDAP enabled, you use your software (client) to use data that is on a OPeNDAP server. Clients include matlab, ferret, GrADS and others. (See http://opendap.org/faq/whatClients.html) OPeNDAP is very convenient; you don't have to worry about grib, netcdf, hdf or binary because everything is converted into an OPeNDAP format/protocol. Using OPeNDAP with GrADS is simple,

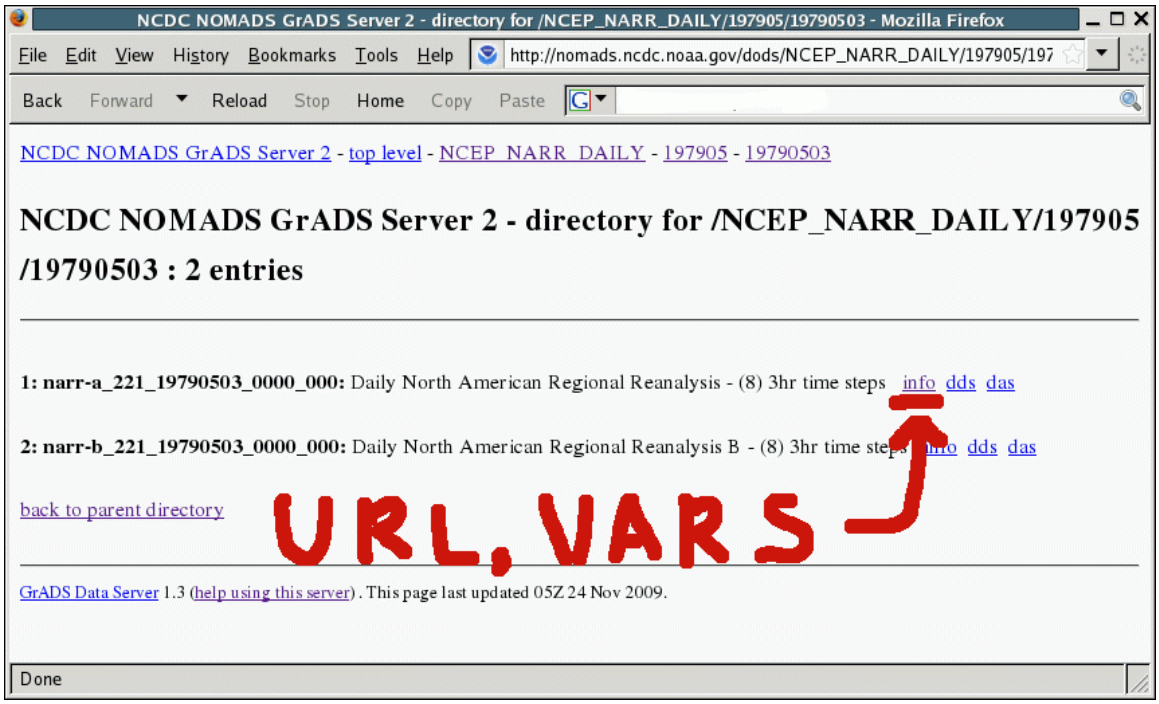
sdfopen URL - each dataset has its own URL

Figuring out the URL involves a little searching. Here is example from the NARR dataset at NCDC.

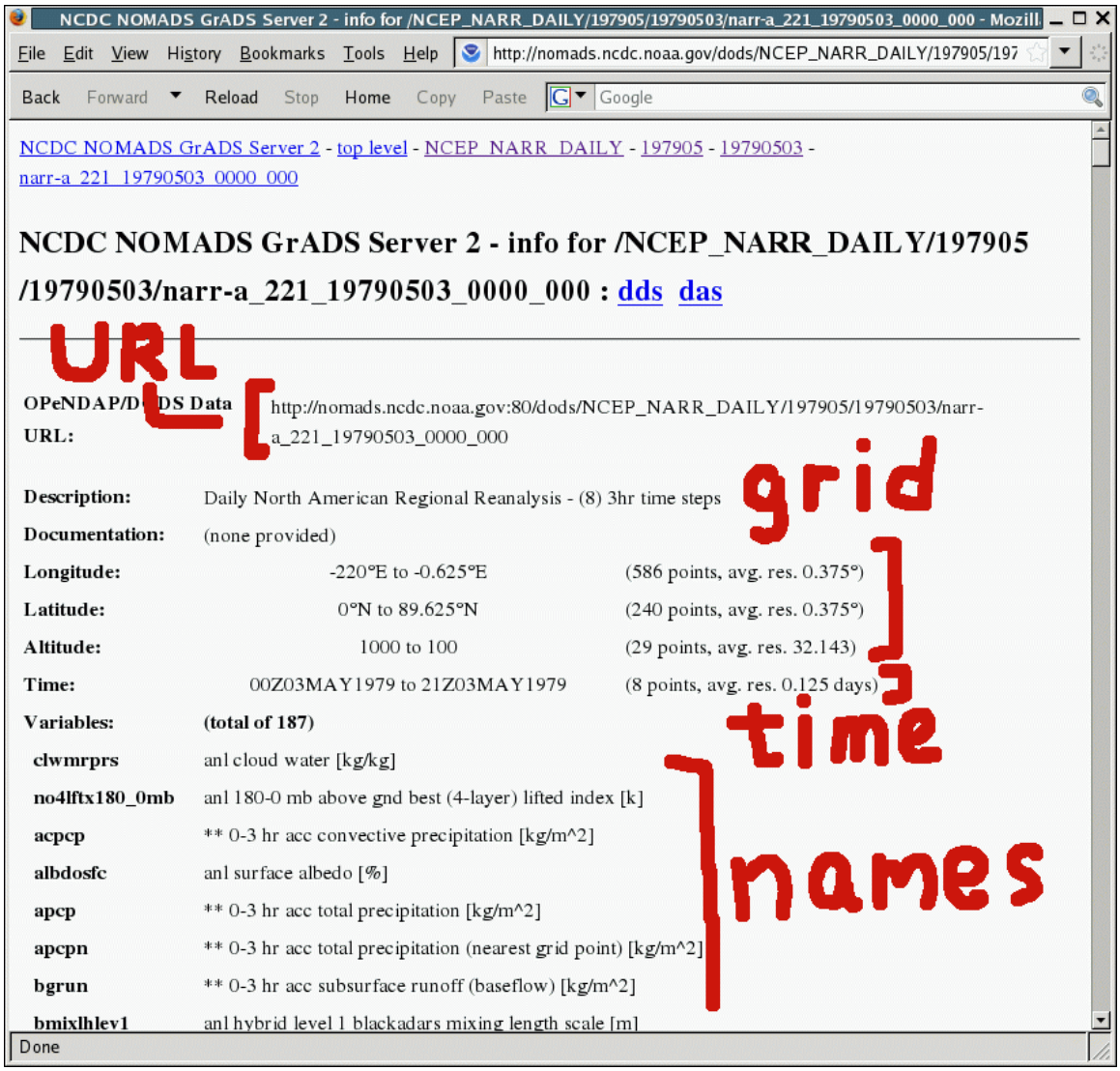
http://nomads.ncdc.noaa.gov/data.php?name=access#narr_datasets



Select a date and you get the following. The “info” link will show the URL (very important), the data set name, grid and variables.



Select the “info” link and you get



The URL is important as your client software will need to know it. The “names” are the variables that are in the file and you will have to scroll down to see if there is a single level or multiple levels, such TMP on the isobaric surfaces.

With the URL, you can run an OPeNDAP-enabled GrADS,

~sh-3.00$ gradsdap

Landscape mode? ('n' for portrait):

ga-> sdfopen http://nomads.ncdc.noaa.gov:80/dods/NCEP_NARR_DAILY/197905/19790503/narr-a_221_19790503_0000_000

ga-> display apcpsfc

GDS is convenient but has some drawbacks.

- data transfer is not compressed like grib-2
- data has to be on latitude-longitude grid (can be irregular in latitude). Polar stereographic and Lambert conformal (NARR) is interpolated to a latitude-longitude by GDS. There can be interpolation errors. For NARR, an oversight caused the regular grid definition to change for newer files at NCDC.
- Much more overhead on the server than the previous methods. The web master is more likely to throttle the high-overhead methods than the more efficient methods.