A DODS Quick Start Guide
Version 1.2

Tom Sgouros

June 29, 2001

Contents

List of Figures

1
What To Do With A DODS URL

The Distributed Oceanographic Data System is a system that allows you to access data over the internet, from programs that weren't originally designed for that purpose, as well as some that were.

With DODS, you access data using a URL, just like a URL you would use to access a web page. However, before you request any data, you need to know how to request it in a form your browser can handle. DODS data is stored in binary form, and by default, it is transmitted that way, too.

The other problem with a DODS URL is that a single URL might point to an archive containing 50 megabytes of data. You rarely want to request the whole thing without knowing a little about it. DODS provides sophisticated sub-sampling capabilities, but you need to know a little bit about the data in order to use them.

Suppose someone gives you a hot tip that there's a lot of good data at:

```
http://www.cdc.noaa.gov/cgi-bin/nph-nc/Datasets/reynolds_sst/sst.mnmean.nc
```

This URL points to monthly means of sea surface temperature, worldwide, compiled by Richard Reynolds at the Climate Modeling branch of NOAA, but pretend you don't know that yet.

The simplest thing you can do with this URL is to download the data it points to. You could feed it to a DODS-enabled data analysis package like Ferret, or you could append `.asc`, and feed the URL to a regular web browser like Netscape. This will work, but you don't really want to do it because in binary form, there are about 28 megabytes of data at that URL.

> **NOTE:** A DODS server will work with many different clients, some of which are supported by the DODS team, and some of which are supported by others. The operation of any individual package is beyond the scope of this manual. This guide explains how to use a typical web browser such as Netscape Navigator to discover information about the data that will be useful when analyzing data in *any* package.

You need to sample the data

A better strategy is to find out some information about the data. DODS has sophisticated methods for subsampling data at a remote site, but you need some information about the data first. First, we'll try looking at the data's *Dataset Descriptor Structure* (DDS). This provides a description of the "shape" of the data, using a vaguely C-like syntax. You get a dataset's DDS by appending `.dds` to the URL.

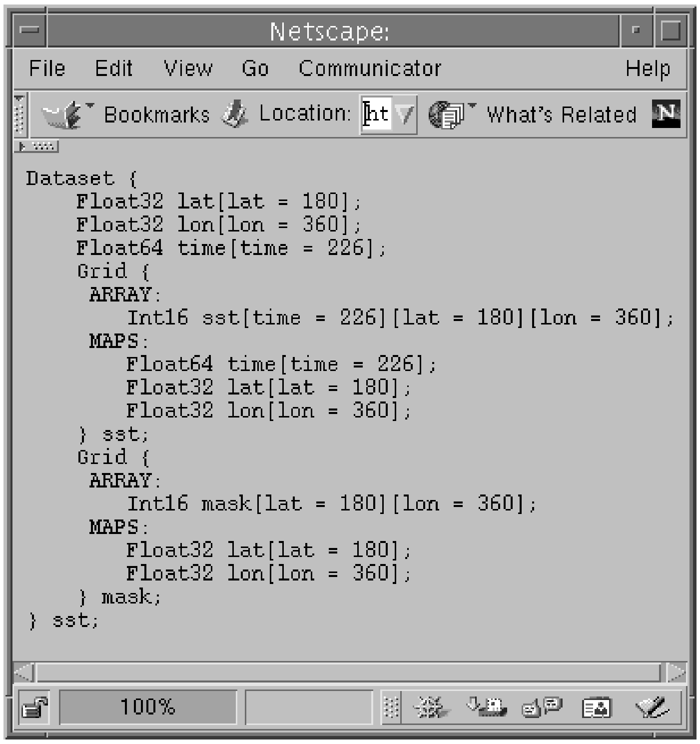

Figure 1.1: A DODS DDS (`sst.mnmean.nc.dds`)

Find out what's in the data

From the DDS shown, you can see that the dataset consists of five pieces:

- A 180-element vector called "lat",
- A 360-element vector called "lon",
- A 226-element vector called "time",

- A “Grid” containing a three-dimensional array of integer values (`Int16`) called `sst`, and three “Map” vectors, which may look familiar, and
- Another Grid called `mask`.

The *Grid* is a special DODS data type that includes a multidimensional array, and *map vectors* that indicate the independent variable values. That is, you can use a Grid to store an array where the rows are not at regular intervals. There’s a simple grid in figure 1.2.

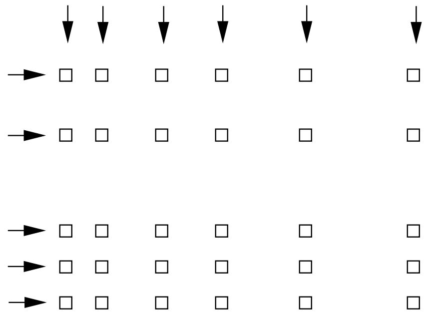

Figure 1.2: A Grid

The array part of the grid would contain the data points measured at each one of the squares, the X map vector would contain the positions of the columns, and the Y map vector would contain the positions of the rows.

Of course you can also use a Grid to store arrays where the columns and rows are at regular intervals, and you’ll often see DODS data that way.

(The other special DODS data type worth worrying about is the *Sequence*. You’ll see more about them in section 1.2. There are also *Structures* and *Lists* , but they exist largely for internal uses, and you don’t often see these used in real datasets.)

You can see from the DDS that the Reynolds data is in a 180x360x226 element grid, and the dimensions of the Grid are called “lat”, “lon”, and “time”. This is suggestive, but not as helpful as one could wish. To find out more about what the data *is*, you can look at the other important DODS structure: the DAS , or *Data Attribute Structure* . This is somewhat similar to the DDS, but contains information about the data, such as units and the name of the variable. Part of the DAS for the Reynolds data we saw above is shown in 1.3.

> **NOTE:** The DAS is populated at the data provider’s discretion. Because of this, the quality of the data in it (the *metadata*) varies widely. The data in the Reynolds dataset used in this example are COARDS compliant. Other metadata standards you may encounter with DODS data are HDF-EOS , EPIC , FGDC , or no metadata at all.

Find out more about the data variables

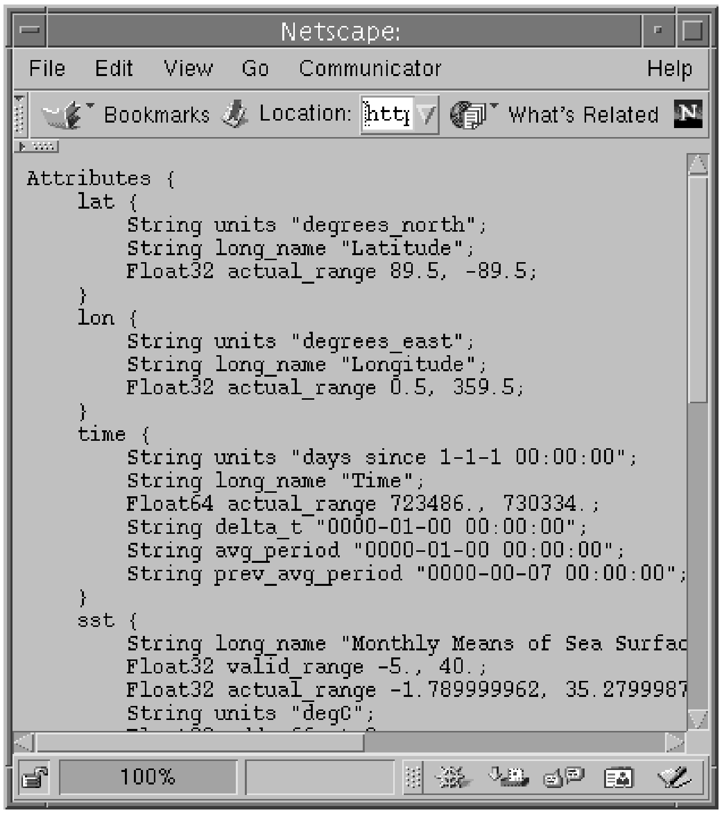

Figure 1.3: A DODS DAS (`sst.mnmean.nc.das`)

Now we can tell something more about the data. Apparently the `lat` vector contains latitude, in degrees north, and the range is from 89.5 to -89.5. Since this is a global grid, the latitude values probably go in order. We can check this by asking for just the latitude vector, like this:

```
http://www.cdc.noaa.gov/cgi-bin/nph-nc/Datasets/reynolds_sst/sst.mnmean.nc.asc?lat
```

What we've done here is to append a *constraint expression* to the DODS URL, to indicate how to constrain our request for data. Constraint expressions can take many forms. This guide will only describe a few of them. (You can refer to the *The DODS User Guide* for more complete information about constraint expressions.) Try requesting the time and longitude vectors to see how this works.

The info service also provides the DAS and DDS information.

According to the DAS, time is kept in "days since 1-1-1 00:00:00" in this dataset. You can also learn from the DAS the actual time period recorded in the data which, because of your familiarity with the Julian calendar , you instantly recognize as beginning in November, 1981. You might also notice that the `mask` array is used to indicate land and sea, and has only the values 0 and 1.

DODS provides an *info service* that returns all the information we've seen so far in a single request. The returned information is also formatted differently (some would say "nicer"), and you can occasionally find server-specific documentation here, as well. Some will find this the easiest way to read the attribute and structure information. You can see what information is available by appending `.info` to a URL, like this:

```
http://www.cdc.noaa.gov/cgi-bin/nph-nc/Datasets/reynolds_sst/sst.mnmean.nc.info
```

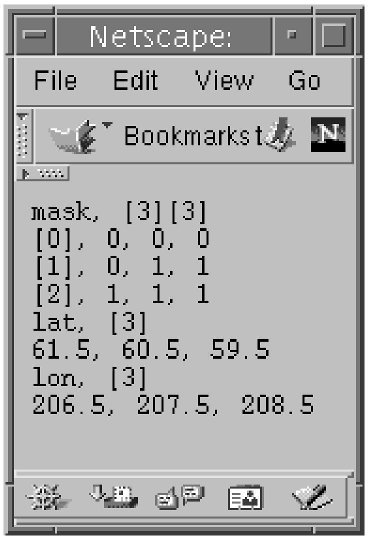

Figure 1.5: Part of a DODS Grid.

1.1 Peeking at Data

Use subscripts to sample a Grid.

Now that we know a little about the shape of the data, and the data attributes, let's look at some of the data.

You can request a piece of an array with subscripts , just like in a C program or in Matlab or many other computer languages. Use a colon to indicate a subscript range.

```
...sst/mnmean.nc.asc?time[0:6]
```

This URL will produce figure 1.4

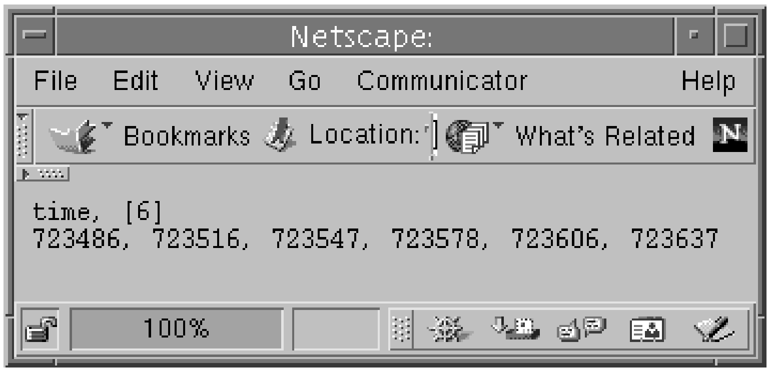

Figure 1.4: Part of a vector.

You can do the same for one of the grids:

Sampling a Grid produces part of the Grid, including the map vectors.

```
...sst/mnmean.nc.asc?mask[28:30][206:209]
```

Which produces a portion of the land mask somewhere near Alaska's Kenai peninsula, shown in figure 1.5

Notice that when you ask for part of a DODS Grid, you get the array part along with the corresponding parts of the map vectors.

If you are interested in the Reynolds dataset, you are probably more interested in the sea surface temperature data than the land mask. The temperature data is a three-dimensional grid. To sample the `sst` Grid, you just add a dimension for time:

```
...sst/mnmean.nc.asc?sst[12:13][28:30][206:209]
```

This produces something like the figure shown in figure 1.6

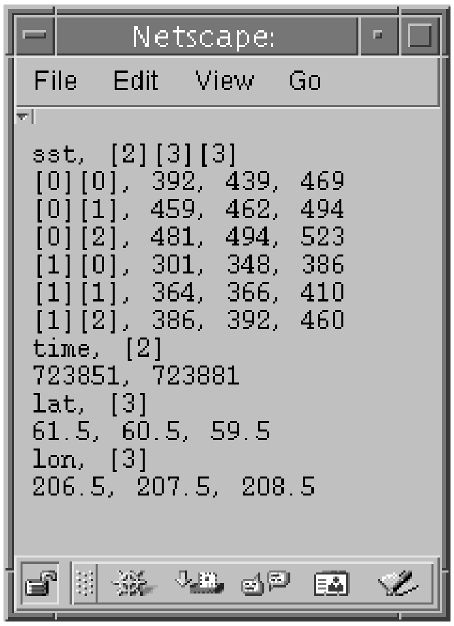

Figure 1.6: Part of the Reynolds SST data

Note that the sst values are in celsius degrees multiplied by 100, as indicated by the `scale_factor` attribute of the DAS. Further, it's important to remember with this dataset, that the data were obtained by calculating spatial and temporal means. Consequently, the data points in the `sst` array should be ignored when the corresponding entry in the `mask` array indicates they are over land.

1.2 Sequence Data

A Sequence is a relational table.

Gridded data works well for satellite images, model data, and data compilations such as the Reynolds data we've just looked at. Other data, such as data measured at a specific site, is not so readily stored in that form. DODS provides a data type called a *Sequence* to store this kind of data.

A Sequence can be thought of as a relational data table , with each column representing a different data value, and each row representing a different data "instance." For example, an ocean temperature profile can be stored as a Sequence of pressure and temperature pairs, and a weather station's data can be stored as a Sequence with time in one column, and each weather variable occupying another column.

Let's look at a couple of Sequences. The first one is a collection of CTD data (hydrographic data , including temperature, pressure, salinity, and so on):

```
http://dods.gso.uri.edu/cgi-bin/nph-jg/rlctd
```

The DAS (append `.das` to the URL) for this data is pretty uninformative, telling us only that all the data are stored as strings. You can see this in figure 1.7.

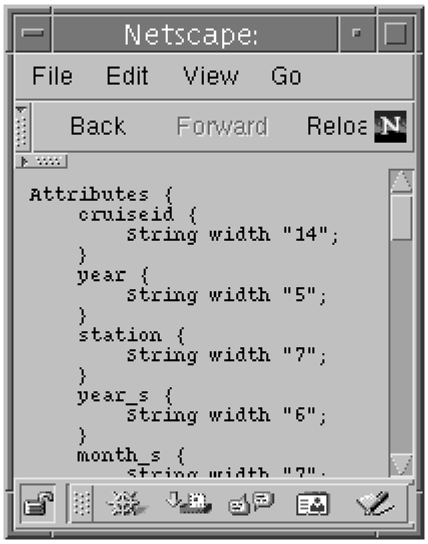

Figure 1.7: A DAS for Sequence data.

You can sometimes find data attributes among the data.

On the other hand, a lot of the information we would get from the DAS is actually encoded in the data itself, which you can see by looking at the data's DDS (append `.dds` to the URL), shown in figure 1.8.

We can get some idea of the data coverage by asking for some of the time and location data, with a URL like this:

```
...rlctd.asc?cruiseid,station,year_s,month_s,day_s,lat_s,lon_s
```

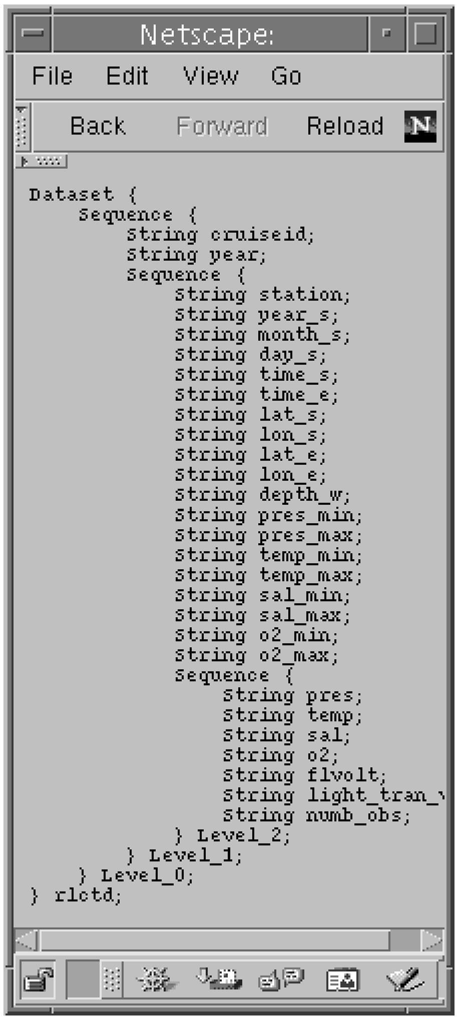

Figure 1.8: A DDS for Sequence data.

Use a selection clause to select Sequence rows.

This produces a response shown in figure 1.9.

After reviewing the data in the last request, perhaps we decide we only want to see data from one of the cruises listed, or maybe only data from the month of May. We can add a *selection clause* to the constraint expression to select only that data. For example:

```
...rlctd.asc?cruiseid,station,year_s,month_s,day_s,lat_s,lon_s&month_s=5
```

This produces a table containing all the rows from figure 1.9 where the month datum is May. Try entering the new URL in your browser and see what you get.

Selection clauses can be stacked endlessly against a URL, allowing all the flexibility most people need to sample data files. Here's an example of a URL that requests all the oxygen data in the file taken in May at a specific depth range:

```
...rlctd.asc?o2&month_s=5&pres>50&pres<100
```

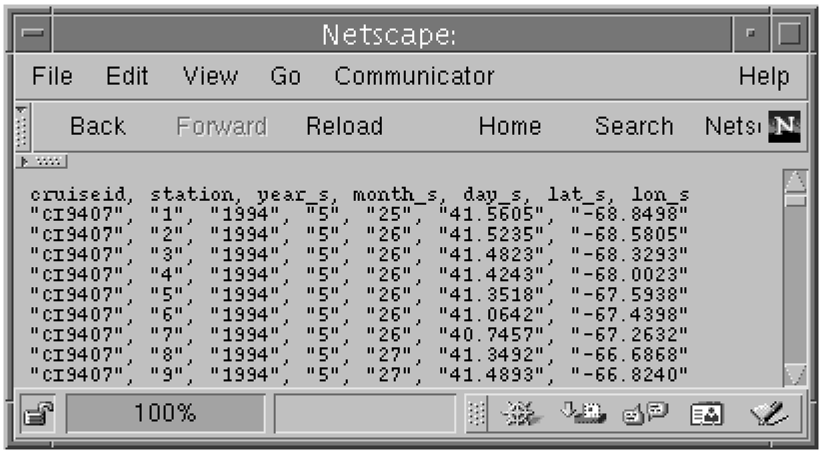

Figure 1.9: The `rlctd` dates and locations

The first clause in a constraint expression has a name, too. It is the *projection clause* . This is the list of variables that you wish to have returned, subject to the constraint of the selection clause. In the previous example, the projection clause consiste only of the `o2` variable. In the one before that, the list was longer, containing 7 variables.

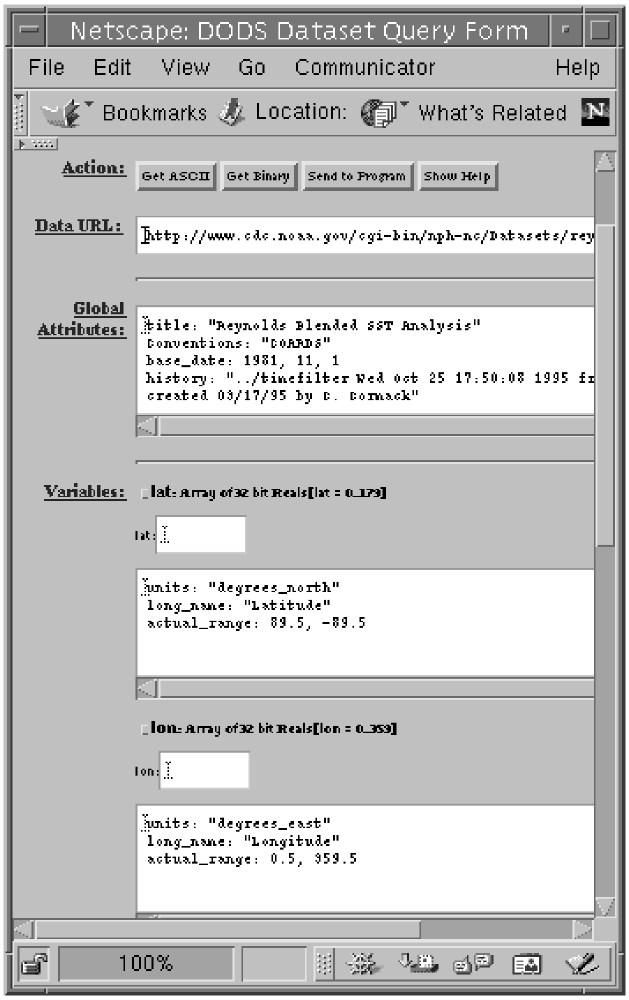

Figure 1.10: The DODS Dataset Access Form

1.3 An Easier Way

The DODS query form is an easier way to sample data.

DODS also includes a way to sample data that makes writing a constraint expression somewhat easier. Append `.html` to the URL, and you get a form that directs you to add information to sample the data at a URL:

```
...sst.mnmean.nc.html
```

Sending a URL ending in `.html` returns a form like this:

It's useful to have a browser window open with one of these query forms in it while you read this section.

Near the top of the page, you'll see a box entitled "Data URL". At this point, if you've been following along, it should look pretty familiar. If you're just jumping in, it's the DODS URL connected to the data we're interested in, but unsampled.

Moving down the page, there is a list of "Global Attributes", which is really just for your perusal. At this point, there's not much to be done with this, but it is often helpful information.

Select variables by clicking on a checkbox.

The important part of the page is the "Variables" section. For each variable in the dataset, you'll see the data description (e.g. "Array of 32 bit Reals [lat = 0..179]"), a checkbox, a text input box, and a list of the variable's attributes. If you click on the checkbox, you'll see the variable's array bounds appear in the text box, and you'll see that variable appear in a constraint expression appended to the Data URL at the top of the page. If you edit the array bounds in the text box, hitting "enter" will place your edits in the Data URL box.

In the oh-so-unlikely event you dare try all this without your documentation *vade mecum* along, there's a **Show Help** button up near the top of the page. Clicking there will show you instructions about how to proceed.

> **NOTE:** You'll see a "stride" mentioned. This is another way to subsample a DODS array or Grid. Asking for `lat[0:4]` gets you the first five members of the `lat` array. Adding a stride value allows you to skip array values. Asking for `lat[0:2:10]` gets you every second array value between 0 and 10: 0, 2, 4, 6, 8, 10.

Move on down the variable list, editing your request, and experiment with adding and changing variable requests.

When you have a request you'd like to make, look at the buttons at the top of the page.

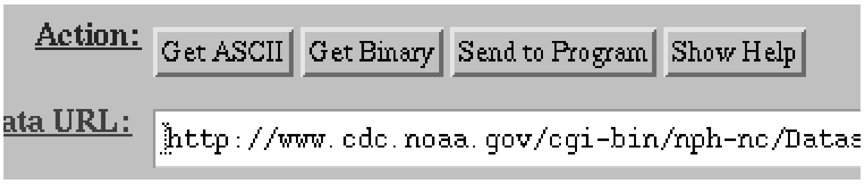

Figure 1.11: Dataset Access Form Detail

You can click on **Get ASCII**, and the data request will appear in a browser window, in comma-separated form. The **Get Binary** button will save a binary data file on your local disk. (The **Send to Program** will send the URL directly to a DODS client. However, it requires a suitable DODS client to be running on your computer, and also requires you to install a helper application for your browser. There are instructions for doing this at the DODS home page.)

The web interface works for Sequence data, too.

The DODS Data Access Form interface works for Sequence data as well as Grids. However, since Sequence constraint expressions look different than Grid expressions, the form looks slightly different, too. You can see from figure 1.12 that the variable selection boxes allow you to enter relational expressions for each variable. Beside that, however, the function is exactly the same.

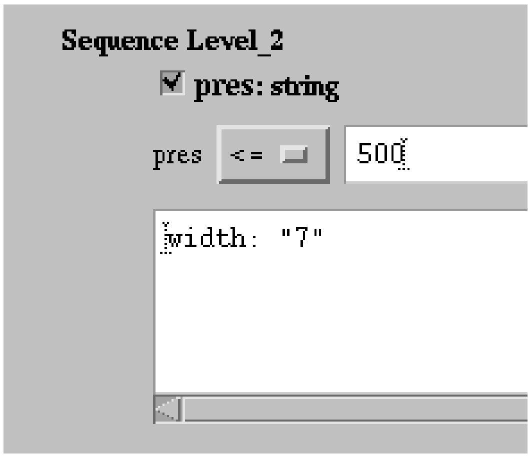

Figure 1.12: Dataset Access Form for Sequence Data (detail)

NOTE: Not all DODS servers support all the DODS functionality. There are a few non-standard DODS servers out there in the world that only support the bare minimum required. That minimum is to respond to queries for the DDS, DAS, and (binary) data. The ASCII data and the web access form are optional add-ons that are not required for the basic DODS function.

2 Finding More DODS URLs

The DODS package was developed to improve ways to share data among scientists. Many times, data comes in the form of a URL enclosed in an email message. But there are several other ways to find data served by DODS servers.

2.1 GCMD

The GCMD now catalogs DODS URLs!

The Global Change Master Directory is a source of a huge amount of earth science data. They now catalog DODS URLs for the datasets that have them. You can search on “DODS” right from the main page to find many of these datasets. Try that search, then click on one of the data set names that returns, and look at the bottom of the resulting Set Description” page, under the heading “Related URL.”

If you make that search, check the list for the Reynolds data from chapter 1; it should be there.

2.2 DODS Dataset List

The DODS Home page has a list of available DODS datasets. Click on **Datasets** in the table of contents. You can find a URL and a brief description for several hundred different datasets from that list.

The DODS project supports an ad hoc list of data URLs.

2.3 Web Interface

The web interface allows browsing data directories.

This is a little bit sneaky. Many sites that serve one DODS dataset serve several others as well. The DODS web interface (if it's enabled by the site) allows you to check the directory structure for other datasets. For example, let's look at the Reynolds data we saw in chapter 1:

```
http://www.cdc.noaa.gov/cgi-bin/nph-nc/Datasets/reynolds_sst/sst.mnmean.nc.html
```

If we use the same URL, but without the file at the end, we can browse the directory of data:

```
http://www.cdc.noaa.gov/cgi-bin/nph-nc/Datasets/reynolds_sst/
```

The DODS server checks to see whether the URL is a directory, and if so, it generates a directory listing, like in figure 2.1.

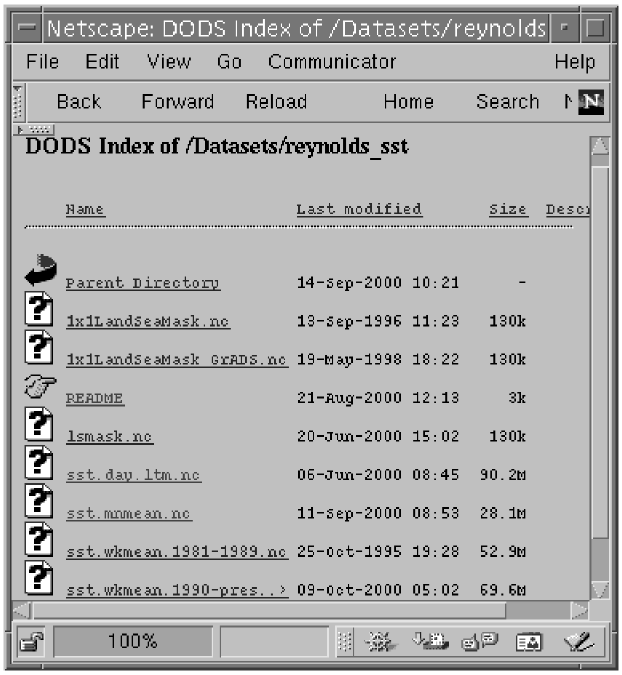

Figure 2.1: Web Interface Index Listing

You can see from the directory listing that the monthly mean dataset we've been looking at is accompanied by a weekly mean set, and a daily set. You can click on those datasets for more information about them, and proceed to examine and use them just as we've done with the other examples in chapter 1.

> **NOTE:** This list is produced by a DODS server. It only really understands DODS data files. If the directory you're looking at has other files in it, clicking on them will probably produce an error.

2.4 File Servers

Some datasets you'll find are actually lists of other datasets. There are a few of these *file servers* in the DODS Dataset List on the DODS Home page. A file server is itself a DODS dataset, organized as a Sequence, containing URLs with some other identifying data (often time). You can request the entire dataset, or subsample it just like any other DODS dataset.

A file server is a list of other datasets, but it's a dataset, too.

There is a file server for GSO/URI's archive of AVHRR sea surface temperature data:

```
http://maewest.gso.uri.edu/cgi-bin/nph-ff/catalog/avhrr.catalog
```

Look at this server's DDS, and the web interface, and then try asking for some data like this:

```
.../catalog/avhrr.catalog.asc?DODS_URL&year=2000&month=1
```

This produces a list of all the data URLs corresponding to measurements taken in the month of January, 2000.

2.5 Matlab GUI

The DODS Matlab GUI browser contains its own frequently updated list of available datasets. Using that software, you can select datasets with a mouse from a large selection of the available URLs. For more information, please refer to the *The DODS Matlab GUI* manual.

The Matlab GUI has its own list of available data.

3 Further analysis

This guide is about forming a DODS URL. After you have figured out how to request the data, there are a variety of things you can do with it. (DODS software mentioned here is available from the DODS Home page.)

Use a generic web client or a DODS client to get the data you've chosen.

- Use a generic web client like `geturl` (a standard part of the DODS package), the free programs `wget` or `lynx`, or even a browser like `Netscape Navigator` or `Internet Explorer` to download data into a local data file. To be able to use the data further, you will probably have to download the ASCII version by using the `.asc` suffix on the URL, as in the examples shown.
- There are pre-packaged DODS clients available that can download binary DODS data from the web into a useful form. As of June 29, 2001, command line clients (`loaddods`) are available for the Matlab and IDL data analysis environments, with which you can download DODS data directly into IDL or Matlab objects.
- The Ferret and GrADS free data analysis packages both support DODS. You can use these for downloading DODS data, and for examining it afterwards. (There are limitations. As of June 29, 2001, Ferret can not read datasets served as Sequence data.)
- The Matlab analysis package also supports a DODS client attached to a graphical user interface. You can use the GUI to create a constrained DODS URL, and download the data directly into Matlab. The *The DODS Matlab GUI* contains more information about the Matlab GUI client.
- If you have a data analysis program or package that you like, you can look into the possibility of linking that package to the DODS toolkit library, in effect making your program into a web-capable DODS client. DODS!libraries exist to mimic the behavior of the NetCDFand JGOFS data

access APIs. If your program already uses one of these APIs, getting it to run with DODS may be as simple as changing the libraries to which you link it. The *The DODS User Guide* describes how to do this, and the *The DODS Toolkit Programmer's Guide* describes how you can use the DODS toolkit directly to create a new application that doesn't use one of the established data access APIs.

The use of these clients, like the ways in which you can analyze the data you find, is beyond the scope of this (or any) book. Enjoy.

Index