The DODS FreeForm ND Server

The DODS FreeForm ND server is a DODS server that uses FreeForm ND to convert and serve data in formats that are not directly supported by DODS servers. FreeForm ND is a flexible system for describing the format of data files and performing file-to-file conversion upon them. Bringing FreeForm ND's data conversion into the DODS world widens data access for DODS clients, since any format that can be described in FreeForm ND can now be served by DODS.

About the DODS FreeForm ND server

Like all DODS servers, the DODS FreeForm ND server responds to client requests for data by returning either data values or information about the data. It differs from other DODS servers because it invokes FreeForm ND to convert the data before serving it to the client.

The following sequence of steps illustrates how the DODS FreeForm ND server works:

1. A DODS client sends a request for data to the DODS FreeForm ND server. The request must include the name of the file that contains the data, and may include a constraint expression to sample the data.
2. The DODS FreeForm ND server looks in its path for two files: a data file with the name sent by the client; and a format definition file to use with the file. The format definition file contains a description of the data format, constructed according to the FreeForm ND syntax.
3. The server uses both files in invoking the FreeForm ND engine. The FreeForm ND engine reads the data file and the format file, using them to convert the data and pass it back to the DODS FreeForm ND server.
4. On receiving the converted data, the DODS FreeForm ND server converts the data into DODS DAP format. The conversion may involve some adjustment of data types; these are listed below. The server also applies any constraint expressions that the client sent along with the URL.
5. The server then constructs DDS and DAS files based on the format of the converted data. If the server has access to DDS and DAS files that describe the data, it applies those definitions before sending them back to the client.
6. Finally, the server sends the DAP, DAS, and converted data to the client.

For detailed information on building and operating the DODS FreeForm ND server, please refer to the INSTALL and README files in the DODS FreeForm ND distribution package.

Getting started with the DODS FreeForm ND server

To get going with the DODS FreeForm ND server, you need to complete these steps:

1. Obtain and install both the core DODS distribution and the DODS FreeForm ND server distribution. (See `http://dods.gso.uri.edu` for more information.)
2. Examine the structure of the data file(s) you intend to use for input, and construct a FreeForm ND format definition file that describes the layout of data in the files. (See the FreeForm ND User's Guide for more information.)

3. If you wish, you may include an output definition format within this file, to allow you to test that your input description is accurate. You can use the FreeForm ND utilities to validate the conversion. (See the FreeForm ND User's Guide for more information.) This step is optional, since the DODS FreeForm ND server ignores the output definition section of the format definition file.
4. Although the DODS FreeForm ND server can generate default DDS and DAS files, you may want to write these files yourself, to override the default data descriptor and attribute structures. (The defaults are based on the format of the data that the DODS FreeForm ND server receives from the FreeForm ND engine).
5. Place the data files, and a corresponding format file for each data file, in a place where the DODS server can find them.
6. Fire up the DODS FreeForm ND server, and use your intended client to request the data. (See the INSTALL and README files for the DODS FreeForm ND server for more information.)

About FreeForm ND

FreeForm ND is a data description and conversion toolkit, originally developed by NOAA's NGDC, that has been adopted for use with DODS.

Rather than define its conversions internally, FreeForm ND uses an external format definition file, which describes the format of the input data and the desired format of the output data. FreeForm ND users write the format definition files, so the data conversion capabilities of FreeForm ND are limited only by the syntax of the format definition, and a user's ability to describe the format of the data. The format definition file can describe data files that are arranged either in tabular form (rows and columns) or array form (elements of variable size). FreeForm ND can read data in ASCII, binary, and DBASE files.

FreeForm ND includes utilities for performing and verifying file-to-file conversions, and libraries for writing applications that make use of its conversion capabilities. See the FreeForm ND User's Guide for a complete description of this software package.

For FreeForm ND users

If you're already familiar with FreeForm ND, you may find these comparisons between the DODS FreeForm ND server and FreeForm ND helpful:

- The DODS FreeForm ND server can be thought of as a FreeForm ND application that converts data on receiving a client request for the data.
- The DODS FreeForm ND server does not produce an output file containing the converted data, but serves it directly to the DODS client. Therefore, the DODS FreeForm ND server ignores the output section of the format definition file.
- To sample a data file, you do not write format definitions that cause the FreeForm ND engine to sample the data file. Instead, you add constraint expressions to the URL that the client sends to the DODS FreeForm ND server.
- The DODS FreeForm ND server performs data conversion on the fly. Conversion only takes place when the client sends a URL requesting data from the DODS FreeForm ND server.

- Unlike FreeForm ND, there is no static file created by the conversion. (If you wish to create or work with such a file, use the FreeForm ND utilities.)

Data type conversions

The DODS FreeForm ND server performs data conversions, based on the data it receives from the FreeForm ND engine.

Basic data type conversions

FreeForm ND	DODS
`text`	`String`
`int8`	`Byte`
`uint8`	`UInt32`
`int16`	`Int32`
`uint16`	`UInt32`
`int32`	`Int32`
`uint32`	`UInt32`
`int64`	`Int32 *`
`uint64`	`UInt32 *`
`float32`	`Float64`
`float64`	`Float64`
`enote**`	`Float64`

* We do not particularly recommend the use of `int64` and `uint64` in the format definition file.

** enote is used for exponential notation

Conversion examples

These examples show how the DODS FreeForm ND server treats data received from the FreeForm ND engine. Please see the DODS FreeForm ND server distribution for more test data and format definition files, and the FreeForm ND User's Guide for more information on writing format definitions.

Arrays

If you define a variable as an array in the format definition file, the DODS FreeForm ND server produces an array of variables with matching types.

For example, this entry in the format definition file:

```
binary_input_data "array"
fvar1 1 4 ARRAY["records" 1 to 101] of int32 0
```

is converted by the DODS FreeForm ND server to:

```
`Int32 fvar1[records = 101]'.
```

Collections of variables

If you define a collection of variables in the format definition file, the DODS FreeForm ND server produces an array of variables with matching types.

For example, this entry in the format definition file:

```
ASCII_input_data "ASCII_data"
fvar1    1 10  int32 2
svar1   13 18  int16 0
usvar1 21 26 uint16 1
lvar1   29 39  int32 0
ulvar1 42 52 uint32 4
```

is converted by the DODS FreeForm ND server to:

```
Sequence {
Int32 fvar1;
         Int32 svar1;
         .
         .
         .
} ASCII_data;
```

Multiple arrays

If you define a collection of arrays in the format definition file, the DODS FreeForm ND server produces a dataset containing arrays.

For example, this entry in the format definition file:

```
binary_input_data "arrays"
fvar1 1 4 ARRAY["records" 1 to 101] of int32 0
fvar2 1 4 ARRAY["records" 1 to 101] of int32 0
```

is converted by the DODS FreeForm ND server to:

```
Dataset {
    Int32 fvar1[records = 101];
    Int32 favr2[records = 101];
    } ... ;
```