

DODS Ancillary Information Service SRS, Draft

James Gallagher*

Revision : 1.16
Printed March 3, 2004

Contents

1	Introduction	3
1.1	Purpose	3
1.2	Scope	3
1.3	Overview	4
2	Overall Description	4
2.1	Product functions	4
2.2	Product perspective	5
2.2.1	System interfaces	5
2.2.1.1	Textual representation of the DAS and DDS objects	5
2.2.1.2	The DODS URL	5
2.2.1.3	Augmenting the DDS object	5
2.2.2	User interfaces	6
2.2.2.1	The DODS URL	6
2.2.2.2	The AIS Server	6
2.2.2.3	The text representation of the DAS and DDS	6
2.2.2.4	Graphical User Interfaces	6
2.2.3	Software interfaces	6
2.2.3.1	DAP 2.0	6
2.2.3.2	HTTP 1.0	6

*The University of Rhode Island, jgallagher@gso.uri.edu

2.2.4	Operations	6
2.2.4.1	User defined AIS files	6
2.2.4.2	Collections of AIS resources should be listed	7
2.2.5	Site adaptation requirements	7
2.3	User characteristics	7
2.4	Constraints	7
2.5	Assumptions and dependencies	8
3	Specific Requirements	8
3.1	External interfaces	8
3.1.1	User interfaces	8
3.1.2	Software interfaces	9
3.1.3	Communications interfaces	9
3.2	System features	9
3.2.1	Specifying AIS resources in the URL	9
3.2.2	Augmenting the DAS	11
3.2.2.1	Writing AIS resources for the DAS object	11
3.2.2.2	Supporting parallel sets of attributes	11
3.2.3	Augmenting the DDS	12
3.2.3.1	Virtual variable values	12
3.2.3.2	Virtual variable value files	12
3.2.4	AIS Servers	14
3.2.5	Groups of data sources	14
3.2.5.1	Selection of specific AIS resources from collections	14
3.2.6	AIS access logging	15
3.3	Performance requirements	15
3.4	Design constraints	15
	References	16
	A Acronyms and Abbreviations	16
	B Change log	17

1 Introduction

This is the Software Requirements Specification for the Distributed Oceanographic Data System Ancillary Information Service.

This document conforms to the IEEE 830-1998 Software Requirements Specification (SRS) recommended practice. Since the recommended practice covers a wide range of possible projects, some of the information in it is not appropriate for this part of DODS. Where that is the case, that section has been marked N/A.

Bold face type is used to indicate a word or phrase which can be found in the glossary.

Emphasized text contained in square brackets ([]) is used to indicate an editorial comment, especially about the information that should be provided in a part of the SRS.

1.1 Purpose

This SRS describes the requirements for the Ancillary Information Service (AIS) and changes in the requirements for other pieces of already existing software.

The intended audience is the entire developer community of the Distributed Oceanographic Data System.

1.2 Scope

The software to be developed is the AIS for DODS. Some existing software will need modification.

The overriding goal of development of the AIS is to achieve level 3 interoperability at the data level in the DODS environment. That is, the AIS will provide semantically meaningful machine-to-machine interoperability for the use of data (as opposed to the discovery of datasets) with distributed, heterogeneous data sets accessed via the DODS DAP

To achieve the goal, the AIS will provide ways for users to add and change the Attributes of a dataset. It will also make it possible for users to add new Variables to a dataset. Finally, the AIS will be designed so that users may add this information directly in a URL which references a dataset or by creating a new virtual dataset using an *AIS server*.

The motivation for the AIS stems from two observations made regarding Distributed Oceanographic Data System (DODS):

1. Complete and consistent syntactic and semantic use metadata are required to achieve Level 3 interoperability.

2. We will not be able to achieve complete and consistent semantic use metadata by adding information at only data server locations.

1.3 Overview

The remainder of this document is organized as follows:

1. Section 2 provides background for the specific requirements and relates those requirements to the rest of DODS.
2. Section 3 lists the specific requirements for the AIS.
3. Following Section 3 are a list of acronyms and abbreviations, a change log, a glossary and references.

2 Overall Description

[This section of the SRS should describe the general factors that affect the product and its requirements. This section should not state specific requirements. Instead, it provides a background for those requirements, which should be defined in Section 3.]

2.1 Product functions

The AIS:

1. MUST provide a way to change for consistency or to add for completeness Attributes (translational use metadata) of or to existing datasets.
2. MUST provide a way to add for completeness (independent) Variables to existing datasets.
3. MUST support making these changes on the computer serving the data, on the client computer or on any other computer on the network.
 - (a) SHOULD support users making those changes without running a network server.
4. Access to an augmented dataset MUST not be logically different from a plain dataset. Clients should be able to use URLs which reference both types of datasets interchangeably.
5. The user MUST be allowed to make use of any combination of AIS DDSs and/or DASs to modify the DDS and/or DAS of a requested data set.

6. There **MUST** be a way to associate public sites providing additions/changes to a given dataset.
7. Legacy clients **MUST** work with AIS URLs, with the possible exception of those URLs which refer to local AIS resources.
8. URLs which contain AIS resources **SHOULD NOT** break legacy servers if at all possible.

2.2 Product perspective

The Ancillary Information Service will meet the goals of Section 1.2 by providing a way to augment the content of the Dataset Attribute Structure (DAS), Dataset Descriptor Structure (DDS) and Data Dataset Descriptor Structure (DataDDS) objects returned by Data Access Protocol (DAP) servers. Together these objects model a data source as a collection of variables, each of which has a name, a datatype, one or more values and a collection of attributes.¹ The AIS provides a way for users to add information to these responses to tailor their content to a particular use.

The following sections describe how the software operates within the constraints imposed by the existing DODS/DAP software.

2.2.1 System interfaces

2.2.1.1 Textual representation of the DAS and DDS objects The text representation of the DAS and DDS objects are described in (Section 2.2.2.3), under the category of *User Interfaces*. However, since these will be used by the software which builds these objects, they are also system interfaces.

2.2.1.2 The DODS Uniform Resource Locator (URL) The DODS URL (see Section 2.2.2.1), like the textual DAS and DDS objects, plays a dual role as both system and user interface. This is true because the DODS software itself uses URLs and because users at various levels “write URLs.”²

2.2.1.3 Augmenting the DDS object The DDS grammar will need modification in order to add virtual variables. These variables will get their values either from other variables or attributes already in the dataset or from external files. The grammar for the DDS object’s external representation will need modification. See Section 3.2.3.

¹See the DAP Specification [2] for a more detailed description of these objects and the other possible requests that can be made to a DAP data source.

²I’m using *users* in a broad sense; it includes people who set up AIS servers as well as people who cut and paste URLs using a web browser.

2.2.2 User interfaces

2.2.2.1 The DODS URL Clients of DODS must be able to specify, in the URL that references a data source, which AIS resource(s) should be used with that data source. This is the most direct way to support all possible interfaces to DODS. This is listed under *User Interfaces* because it is possible that users will type these into a web browser, script file, *et cetera*.

2.2.2.2 The AIS Server The *AIS Server* is a DODS server that provides access to a virtual dataset which includes AIS resources. This server provides a way for people to use the AIS without having to explicitly embed AIS resource references in URLs. The AIS Server URL is a regular DODS URL; the AIS Server hides the additional complexity of explicit AIS resource references.

2.2.2.3 The text representation of the DAS and DDS Information will be provided to the AIS using text documents. These documents may be files or they may be the result of dereferencing a URL. The DAP uses a textual representation for the external form of the DAS and DDS objects. This representation will be used to supply the extra information about specific instances of these objects that constitute the information accessed through the AIS.

2.2.2.4 Graphical User Interfaces Various graphical user interfaces which have been developed for DODS or have been modified to take advantage of DODS may need modification to take full advantage of the AIS. This will depend on the particular GUI in question, describing how each should be enhanced is beyond the scope of this SRS.

2.2.3 Software interfaces

2.2.3.1 DAP 2.0 The DAP protocol defines the DAS, DDS and DataDDS objects.

2.2.3.2 HTTP 1.0 DODS clients and servers use HyperText Transfer Protocol (HTTP) for all their network I/O.

2.2.4 Operations

2.2.4.1 User defined AIS files Users may write their own AIS resources. This will require resources from *User Support*.

2.2.4.2 Collections of AIS resources should be listed Data users (people principally interested in accessing data using the DAP) will need to be able to locate AIS resources or AIS servers. It seems likely that various groups or individuals will create AIS resources for the groups of data sources in which they are interested. Ancillary Information Service resource collections should be listed, if their authors agree to it, so that other users may find them.

2.2.5 Site adaptation requirements

Users will need to upgrade their clients to use local AIS resources.

Providers may need to upgrade their servers.

2.3 User characteristics

There is a range of possible users of the AIS. One group of user will be involved in writing the AIS resources. They will be looking at data sources, seeing what information they contain and thinking about the types of additional information that would make the data source more broadly useful. In many cases these users will be people using the DAP to get data. They will have a good idea of the type of information that needs to be added to the data source to make its contents more useful to them.

Some AIS users will be involved in adding functions to servers so that derived variables can be accessed. These users are essentially an extension of the programmers who will implement this SRS and will be the most technical group of users.

Another type of user will be solely interested in accessing data and will never create their own AIS resources. Some of these users will be aware of the AIS because they are writing URLs that appear in scripts or in configuration files for analysis programs. Others may access data through web systems that hide the use of the AIS.³

Most users of DODS will not want to write AIS resources. Writing this type of information so that various standards and conventions are satisfied is tedious.

2.4 Constraints

[This section lists any other items that may limit the developer's options.]

Performance should be comparable to plain access. Response time increase should be minimal.

³Note that even though some users might not know they are using the AIS, its use can be traced by going back to client programs and looking at the URLs, which will contain the references to any AIS resources used or by getting information from an AIS Server about the AIS resources it uses.

Changes to the current URL syntax should be limited. Access to a DAS object currently does not use a Constraint Expression (CE), however, the AIS may add this. The ‘function call’ syntax already present in the DODS URL’s syntax should be used; another syntax should not be added. It’s OK if the AIS adds a special function which both clients and servers can look for.

The AIS should use as much of the WWW/HTTP infrastructure as possible, building its own infrastructure only when necessary. This will lead to a system that is first and foremost ‘document based,’ which will make deployment and testing simpler.

2.5 Assumptions and dependencies

The system designers can assume that the following are true:

1. People providing AIS resources for use by others expect to install some sort of network server software (e.g., httpd) or to upgrade existing servers.
2. The DAP specification (version 2.0) will be changed to allow AIS resources to be referenced in a DODS URL.
3. A new server will be written to handle the AIS Server operations.

3 Specific Requirements

[This section of the SRS lists all of the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the systems satisfies those requirements.]

This section is organized using template A.5 of IEEE STD 830-1998, organization by feature.

3.1 External interfaces

3.1.1 User interfaces

Each of the following user interfaces have been discussed previously in Section 2.2.2. Note that these are also system interfaces.

1. The DODS URL [2].
2. The text representation of the DAS and DDS objects [2].
3. The AIS resource collection configuration file for remote resources.

3.1.2 Software interfaces

Software to process DODS URLs is located on both the client- and server-side of the system in object-oriented class libraries. Two classes, one each for clients and servers, will be principally involved in processing AIS references embedded in a URL.

1. **Connect**. The `Connect` class handles the task of dereferencing a DODS URL and processing the response, creating other classes and invoking their methods as needed. The implementation of this class will require modification for local AIS resource access.
2. **DODSFilter**. The `DODSFilter` class processes the request for a given object. The class uses information passed from the `DODS_Dispatch.pm` Perl module to build an object and generate the response using that object. This object will require modification for server-side AIS resource access.

In addition, the DAS and DDS classes contain an interface that provides a way for information contained in a text file to be merged with an existing instance. Each class supports a `parse(...)` method which can be invoked with either a pointer to an open file or the name of a file. The file's contents will be merged with the object if the contents parse as a valid object of the corresponding type. That is, if `dds` is an instance of DDS, calling `dds.parse("myfile")` will merge into `dds` the information in `myfile` if its contents are a valid DDS object description using the text notation described in the DAP specification, version 2.0 [2].

DODS servers interface with HTTP daemons using Common Gateway Interface (CGI) 1.1 [3]. The specification describes how different parts of a URL are broken up and passed to CGI programs.

3.1.3 Communications interfaces

DODS client programs use HTTP 1.0 to communicate with server programs.

3.2 System features

3.2.1 Specifying AIS resources in the URL

To request that the object (DAS, DDS or DataDDS) returned from a data source be augmented using information from a particular AIS resource, that resource must be identified in the DODS URL requesting the object.⁴ To do this the syntax of a DODS URL will be modified so that the `query` component can contain both a CE and a list of AIS resources. In the DAP 2.0 specification,

⁴This scheme, of passing the AIS resource using the URL, was suggested by Steve Hankin.

the query part of the DODS URL contained only the CE. To accommodate one or more AIS references it becomes:

```

query = CE | AIS | CE "&" AIS
AIS   = "ais(" AIS_REF *(", " AIS_REF) ")"

```

The complete grammar is:

```

DAP_URL      = "http:" "://" host [ ":" port ] [ abs_path ]
abs_path     = server_path dataset_id "." ext [ "?" query ]
server_path  =
dataset_id   =
ext          = "das" | "dds" | "dods" | "ver" | "html" | "info"
              | "asc"

query        = CE | AIS | CE "&" AIS
AIS          = "ais(" AIS_REF *(, AIS_REF) ")"
AIS_REF     = *(merge) AIS_URL
merge       = "default" | "replace" | "fallback"
AIS_URL     = "http:" "://" host [ ":" port ] [ ais_path ]
              | "file:" "://" [ ais_path ]
ais_path    = "/" | 1*("/" <word>) ".ais"a

CE          = *(projection) *(& selection)
projection  = ids | function
ids         = id | id , ids
function    = id ( args )
args       = arg | args arg
arg        = id | <quoted string> | <int> | <float>
              | [ deref ] URL

```

^aSee Section 3.2.5 for information about AIS resource references that end in a slash.

This syntax will correctly parse both URLs which reference AIS resources as well as older URLs which may have already been embedded in application configuration files, web sites, *et cetera*.

The AIS URL is a regular URL as described by RFC 2396, “Uniform Resource Identifiers (URL): Generic Syntax” [1]. In practice remote resources will only be accessed using the HTTP, HTTPS or FILE protocols, but this issue should be handled by the underlying network I/O layer, not the DODS or DAP software. For example, `libwww` supports a variety of transport protocols. The ‘documents’⁵ referenced will have the extension `.ais`. Information that augments a DAS and/or a DDS may appear in an `.ais` document. The textual representations for these will be used and their current form allows both to be int

⁵The information returned by dereferencing the AIS URL may be held in a file, a database, generated by a CGI, *et cetera*.

the same file unambiguously (see the DAP 2.0 Specification). Note that the URL may describe a remote resource accessed via HTTP, *et c.*, or a local resource accessed using the `file:` protocol identifier.

Example 1:

```
http://www.dods.org/dods-3.2/nph-dods/data/fnoc1.nc.das?
  &ais(http://unidata.ucar.edu/dods/fnoc1_fgdc.adas)
```

In Example 1 the DAS object for the FNOC1 data set is accessed and merged with the extra DAS information in the AIS resource referenced by `fnoc1_fgdc.adas` held at Unidata.

When a user requests a DAS object, the only permissible AIS resource is one that augments a DAS object. See Section 3.2.3 for a discussion of DDS objects affected by attribute (DAS) information.

3.2.2 Augmenting the DAS

3.2.2.1 Writing AIS resources for the DAS object An AIS resource that augments a DAS object is a text file that contains structured text using the grammar for the external representation of a DAS object as described in “DODS DAP 2.0-Draft” [2].

Example 2:

```
Attributes {
  u {
    String long_name "U-component of the vector wind";
  }
}
```

3.2.2.2 Supporting parallel sets of attributes The AIS will not extend the current DAS syntax to support parallel sets of attributes. Instead AIS resource authors will have to write separate AIS resources for each set of attributes.

If a URL writer includes two or more AIS resources, and both provide an attribute with the same name but a different value, the AIS will behave as follows:

1. The resource listed first will take precedence; its value will be the value of the attribute. This is the default behavior; no special notation is needed.
2. Attributes are replaced. Only the attributes from the AIS resource are used. This is called ‘replacement behavior;’ to get it the AIS URL must be prefixed by `replace`.
3. Attributes from the AIS resource are used if and only if they are not defined in the data source. This is called ‘fallback behavior;’ to get it the AIS URL must be prefixed by `fallback`.

3.2.3 Augmenting the DDS

The AIS DDS will contain the textual representation of a DDS as described in the DAP 2.0 specification [2] as amended by this document.

AIS resources which augment a data source's DDS object will add new variables to a data source. Client programs will be able to request these variables just as they can request the 'real' variables.

To denote that a variable has been added to a data source by an AIS resource, that variable's declaration will be prefixed by the type modifier `virtual`. For the sake of convenience these variables will be referred to as **virtual variables**.

3.2.3.1 Virtual variable values While there may be other sources of virtual variables supported in the future, in this version of the AIS SRS only values read from files or other variables or attributes already present in the dataset will be allowed.

When variables or attributes are used as a source of values for virtual variables, their names should be prefixed by, `'Dataset.'` and `'Attributes.'`, respectively.⁶

When the values for a virtual variable are read from a file, the name of the file is given using a function-like syntax. This syntax is used to simplify future expansion to other sources for values.

3.2.3.2 Virtual variable value files The files which hold the values for a virtual variable will use a CSV format. Most virtual variables will be either arrays of simple-types or additional columns in Sequences. The format of the virtual variable file will only address adding values for those datatypes. In the future this may be expanded.

1. A `#` in the file will start a comment which will run to the end of the line.
2. Values in the file will be either numbers (using the same syntax for integer and floating values as is used by the DAS) or strings. Strings which contain spaces must be quoted using double quotes, otherwise the quotes are optional.
3. For one and two dimensional arrays of simple-types, each row of the file will hold the values for a row of the array. Rows will be delimited by a carriage return (`'\n'`). Values for each column position will be separated by commas (`','`).
4. For N -dimensional arrays of simple-types, dimensions $3, \dots, N$ will be represented by repeating the pattern for dimensions $2, \dots, N-1$ separated by a blank line. (Note that the reader will know the dimensionality of the data because that information is in the DDS.)

⁶This extends the concept that these are container objects; it seems like a syntactically natural way to unambiguously denote the source of the values.

5. For a column in a Sequence, each value will appear on a separate row.

The Example below shows how the virtual variable feature can be used to transform a simple array (Called SST in the data source) into a Grid.⁷ The Grid is called `sea_surface_temperature` and its array part is a virtual variable called SST which gets its value from the data source's `temp` variable. The two map vectors for the `sea_surface_temperature` Grid are also virtual variables; their values read from files.

Example:

```
Dataset {
  virtual Grid {
    Array:
      virtual Byte SST[longitude = 512][latitude = 512] = Dataset.temp;
    Maps:
      virtual Float64 longitude[512] = file("long.dat");
      virtual Float64 latitude[512] = file("lat.dat");
  } sea_surface_temperature;
}
```

A partial grammar, updated to include specification of values for virtual variables, for the DDS object's external representation is shown below. This grammar does not show the keyword's various literal representations and a few other details that are not relevant to virtual variables. For the complete grammar see [2].

```
datasets      = +(dataset)
dataset       = dataset "{" *(decl | v_decl) "}" name ";"
decl          = list decl
              | base-type var ";"
              | structure "{" *(declaration) "}" var ";"
              | sequence "{" *(declaration) "}" var ";"
              | grid "{" array : *(declaration)
                maps : *(declaration) "}" var ";"
v_decl        = base-type var "=" value ";"a
base-type     = byte | int16 | uint16 | int32 | uint32
              | float32 | float64 | string | url
var           = id | var array-decl
value         = "Dataset." nameb | "Attributes." namec
              | "file(" string ")"
array-decl    = "[" integer "]" | "[" id "=" integer "]"
```

^aEven though the grammar allows any base-type (both scalars and vectors) initial versions of the AIS will limit these to arrays and scalars within Sequence variables.

^b*name* is the name of a variable as it would appear in a CE.

^c*name* is the fully-qualified name of an attribute.

⁷See the DAP 2.0 specification [2] for an explanation of data types supported by the DODS DAP.

3.2.4 AIS Servers

The AIS Server provides a way to hide the use of AIS resources behind an interface that acts like a regular DODS server.

The AIS Server provides access to one or more DODS datasets using regular DODS URLs.

Each dataset to which an AIS Server provides access is really served by another DODS server (e.g., the DODS-HDF4 server).⁸ A configuration file provides the AIS Server with the information it needs to match URLs to specific AIS resources.

3.2.5 Groups of data sources

[Remove the idea of groups; make this about sites. Allow for RDBs to maintain the list of things. That is, redefine in the guise of a service. Add a note about REs when using a file.]

It will be possible to use a single AIS resource token with a group of data sources. For example, a collection of FNOC data sources might have a matching collection of AIS resources. As a convenience to clients, it will be possible to *reference the collection* of AIS resources in the DODS URL in place of a specific AIS resource reference. The AIS will choose the appropriate resource from the collection for a particular data source. This will simplify using the AIS since users can name the collection and let the AIS figure out which parts of the collection apply to the given data source.

To indicate that a particular reference is to a collection of AIS resources, the reference must end in a slash (/). See Example 1.

Example 1:

```
http://www.dods.org/dods-3.2/nph-dods/data/fnoc17.nc.das?  
&ais(http://unidata.ucar.edu/fnoc_fdgc/)
```

Ancillary Information Service resource collections may be either local or remote. Note that a remote resource collection must refer only to resources local to itself due to the prohibition against chaining. *[Is this really true?]*

3.2.5.1 Selection of specific AIS resources from collections Whether local or remote, AIS resource collections must contain a mapping of the elements of the collection to individual data sources. For local collections, this mapping will be held in the local configuration file. Remote resources will use a configuration file at the same location as the resources themselves. In both cases the same syntax will be used.

[Remove this example; add grammar; add better examples.]

⁸Think of an AIS Server as a kind of proxy server.

Example:

```
fnoc_fdgc {
    http://www.dods.org/dods-3.2/nph-dods/data/fnoc1.nc.das =
        http://unidata.ucar.edu/fnoc_fdgc/fnoc1_fgdc.ais;
    .
    .
    .
    http://www.dods.org/dods-3.2/nph-dods/data/fnoc17.nc.das =
        http://unidata.ucar.edu/fnoc_fdgc/fnoc17_fgdc.ais;
    .
    .
    .
}
```

[Grammar here.]

3.2.6 AIS access logging

AIS accesses at a site will be recorded. This means that accesses made using the 'AIS resource in the URL' and through an AIS server will both be recorded.

AIS Servers will provide access to the logfile (as an ASCII text file) using the special URL `http://< host >/< cgi_bin >/nph_dods/ais_log_info`. Sites that host an AIS server will be able to disable this feature.

AIS resource references embedded in a URL will be logged by the machine that serves those resources. It will be possible to request AIS resource logging information using the same URL as for the AIS Server. Sites will be able to disable this feature.

In both cases accesses shown in the `ais_log_info` document will reflect only accesses to AIS resources.

3.3 Performance requirements

N/A

3.4 Design constraints

Our use of the query string in a URL flies in the face of RFC 2396 [1]. This is potentially a problem although so far no servers have failed because of the technically illegal characters we use. If we were to require strict adherence to the RFC, then many characters in the query string portion of the URL would have to be escaped according to RFC 2396 [1]. This make writing URLs by hand virtually impossible and greatly hinders reading them, even by programmers, as

well. For these reasons, I decided to relax the constraints on characters describe by RFC 2396 for DODS constraint expressions. We'll do the same here.⁹

References

- [1] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform resource identifiers (URI): Generic syntax. RFC 2396.
- [2] James Gallagher. DODS DAP 2.0 – draft.
<http://unidata.ucar.edu/packages/dods/design/dap-rfc-html/>.
- [3] W3C. Cgi: Common gateway interface. <http://www.w3.org/CGI/>, 1999.

A Acronyms and Abbreviations

AIS Ancillary Information Service

BNF Backus-Naur Form

CE Constraint Expression

CGI Common Gateway Interface

DAP Data Access Protocol

DAS Dataset Attribute Structure

DDS Dataset Descriptor Structure

DODS Distributed Oceanographic Data System, See the DODS home page:
<http://unidata.ucar.edu/packages/dods/>

DataDDS Data Dataset Descriptor Structure

HTML Hypertext Markup Language

HTTP HyperText Transfer Protocol

MIME Multimedia Internet Mail Extensions

SRS Software Requirements Specification, See IEEE 830–1998

URI Uniform Resource Identifiers

URL Uniform Resource Locator

W3C The World Wide Web Consortium, See <http://www.w3c.org/>

⁹This does not mean I think all users should be writing URLs, just that I think we should spend some effort to minimize the unpleasantness of that experience.

XML Extensible Markup Language

FGDC Federal Geographic Data Community

B Change log

\$Log: ancillary_information.tex,v \$

Revision 1.16 2004/03/03 17:22:48 jimg

Needs to be looked at... I changed the date to fix an error with CVS. This needs some updating.

Revision 1.15 2003/10/15 15:26:29 jimg

check point

Revision 1.14 2002/01/03 14:34:35 jimg

Most of section three is now complete.

Revision 1.13 2002/01/01 03:10:55 jimg

Sections 1 and 2 are mostly redone.

Revision 1.12 2002/01/01 01:27:14 jimg

I removed much of the old text that had been moved to comments (mostly about tokens).

Revision 1.11 2002/01/01 01:24:29 jimg

Added information from the ais_goals.txt document.

Revision 1.10 2001/11/30 23:50:26 jimg

This version of the paper is still incomplete WRT changes that have been suggested. However, a new architecture has been proposed for the AIS and it seems like that should be investigated before more time is spent refining the ideas here.

Revision 1.9 2001/11/14 00:37:56 jimg

Fixed a bunch of spelling errors.

Revision 1.8 2001/11/13 07:19:27 jimg

Spelling & Grammar fixes.

Revision 1.7 2001/11/11 20:23:26 jimg

Added information about resource collections (a way to reference an AIS site and have it choose the correct resource for a particular file/dataset). Also added more information about AIS resources for the DDS object. Finally, added a constraint that chaining not be allowed (although this is somewhat vague to me).

Revision 1.6 2001/11/08 00:41:55 jimg
Added a description of the way variables will be added to datasets based on the discussions on DODS-tech@unidata.ucar.edu.
Still to go: Prohibition against chaining (?), Mappings at the AIS site(s), Mappings at client, Translation (NB: Per conversation with Dan, maybe this is a separate beast?).

Revision 1.5 2001/11/03 05:26:24 jimg
Changes from Peter and Paul. More changes to come. See the changebars.

Revision 1.4 2001/10/23 05:39:45 jimg
Fixed up section three. Added outline to DDS part.

Revision 1.3 2001/10/20 00:07:28 jimg
Added first crack at Section 3.
Still to do: Good examples, DDS.

Revision 1.2 2001/10/18 23:30:10 jimg
Completed first go at section two.

Revision 1.1 2001/10/18 21:22:02 jimg
First crack at the introduction.