Appendices

Appendix A: NCSA HDF Tags

Overview

This appendix includes tables containing brief descriptions of most of the tags that have been assigned at NCSA for general use. This list will be expanded in future editions to include new tags as they are assigned. A more detailed description of the tags can be found in the *HDF Specifications* manual.

Each table contains a list of tags within one category. The titles of the tables, with a functional description of each table, are:

- **Table A: The HDF Utility Tags.** Used by the HDF utilties.
- **Table B: The HDF General Raster Image Tags.** Used to describe aspects of raster image data.
- **Table C: The HDF Composite Image Tags.** Used to describe aspects of composite image data.
- **Table D: The HDF Scientific Data Set Tags:** Used to describe aspects of scientific data set (SDS) data.
- **Table E: The HDF Vset Tags.** Used to describe aspects of HDF Vset data.
- **Table F: The Obsolete HDF Tags:** Used to describe aspects of HDF data elements that have been replaced by newer tags or discontinued.

Tag Types and Descriptions

The following tables have five columns: the "Tag Name" column contains the abbreviated symbolic names of tags that are often used in an augmented form in HDF programs, the "Short Description" column contains a brief (four word maximum) description of the tag that is commonly used to describe to the tag in HDF manuals and in-line code documentation, the "Data Size" column describes the type of data that is associated with the tag and, where possible, lists the data size, the "Number" column list the numeric value of the tag symbol in the "hdf.h" header file, and the "Long Description" column contains a general description of the tag. The "Data Type" column contains a generalized description of the type of data referred to by the tag.

In the tables, any entry specified as "String" refers to a sequence of ASCII characters with the null byte possibly occurring at the end, but nowhere else. Any entry specified as "Text" also refers to a sequence of ASCII characters, but it may contain zero bytes anywhere in the sequence. An entry in the "Data Type" specified by an "n" refers to a data unit of variable-length. For a more detailed description of these units of data, refer to the appropriate tag entry in the *HDF Specification Manual*.

TABLE A

The HDF Utility Tags

Tag Name	Short Description	Data Size	Tag Value	Long Description
DFTAG_NULL	No Data	None	001	Used for place holding and filling up empty portions of the Data Descriptor Block.
DFTAG_VERSION	Library Version Number	4 bytes + string	030	Specifies the latest version of the HDF library used to write to the file.
DFTAG_NT	Number Type	4 bytes	106	Used by any other element in the file to specifically indicate what a numeric value looks like.
DFTAG_MT	Machine Type	0 bytes	107	Specifies that all unconstrained or partially constrained values in this HDF file are of the default type for that hardware.
DFTAG_FID	File Identifier	String	100	Points to a string that the user wants to associate with this file. This supports the inclusion of a user-supplied title for the file.
DFTAG_FD	File Descriptor	Text	101	Points to a block of text describing the overall file contents. It is intended to be user-supplied comments about the file.
DFTAG_TID	Tag Identifier	String	102	Provides a way to determine the meaning of a tag stored in the file.
DFTAG_TD	Tag Descriptor	Text	103	Similar to DFTAG_TD, but allows more text to be included.
DFTAG_DIL	Data Identifier Label	String	104	Associates the string with the Data Identifier as a label for whatever the identifier points to. By including DILs, any data element can be given a label for future reference. For example, this tag is often used to give titles to raster image data sets.
DFTAG_DIA	Data Identifier Annotation	Text	105	Associates the text block with the Data Identifier as an annotation for whatever that Data Identifier points to. With DIAs, and Data Identifier can have a lengthy, user-provided description of why that particular data element is in the file.
DFTAG_RLE	Run-length Encoding	0 bytes	011	Specifies that run-length encoding (RLE) is used to compress a raster image.
DFTAG_IMC	IMCOMP Compression	0 bytes	012	Specifies that IMCOMP compression is used to compress a raster image.
DFTAG_JPEG	24-bit JPEG Compression	n bytes	013	Provides header information for 24-bit JPEG-compressed raster images.
DFTAG_GREYPEG	8-bit JPEG Compression	n bytes	014	Provides header information for 8-bit JPEG-compressed raster images.

TABLE B

The HDF General Raster Image Tags

Tag Name	Short Description	Data Size	Tag Value	Long Description
DFTAG_RIG	Raster Image Group	n*4 bytes	306	Lists the Data Identifiers (tag/reference number pairs) that uniquely describe a raster image set.
DFTAG_ID	Image Dimension	20 bytes	300	Defines the dimensions of the two-dimensional array the corresponding RI tag refers to.
DFTAG_LD	LUT Dimension	20 bytes	307	Defines the dimensions of the two-dimensional array the corresponding LUT tag refers to.
DFTAG_MD	Matte Dimension	20 bytes	308	Defines the dimensions of the two-dimensional array the corresponding MA tag refers to.
DFTAG_RI	Raster Image	x*y bytes	302	Points to a raster image data set.
DFTAG_CI	Compressed Image	n bytes	303	Points to a compressed raster image data set.
DFTAG_LUT	Lookup Table	n bytes	301	Table to be used by the hardware for the purpose of assigning RGB or HSV colors to data values.
DFTAG_MA	Matte Data	n bytes	309	Points to matte data.
DFTAG_CCN	Color Correction	n bytes	310	Specifies the gamma correction for the raster image and color primaries used in the generation of the image.
DFTAG_CFM	Color Format	String	311	Indicates the interpretation to be given to each element of each pixel in a raster image.
DFTAG_AR	Aspect Ratio	4 bytes	312	Indicates the aspect ratio of the image.
DFTAG_XYP	XY Position	8 bytes	500	Specifies the screen X-Y coordinate for raster image sets. (Also used for composite image sets - See the entry for DFTAG_XYP in Table 12.6)

TABLE C

The HDF Composite Image Tags

Tag Name	Short Description	Data Size	Tag Value	Long Description
DFTAG_DRAW	Draw	n*4 bytes	400	Specifies a list of Data Identifiers (tag/reference number pairs) which define a composite image.
DFTAG_XYP	XY Position	8 bytes	500	Specifies the screen X-Y coordinate for composite image sets. (Also used for raster image sets - See the entry for DFTAG_XYP in Table 12.5)
DFTAG_RUN	Run	n bytes	401	Identifies code that is to be executes as a program or script.
DFTAG_T14	Tektronix 4014	n bytes	602	**Used as a vector image tag.** Points to a Tektronix 4014 data. The bytes in the data field, when read and sent to a Tektronix 4014 terminal, will be displayed as a vector image.
DFTAG_T10S	Tektronix 4015	n bytes	603	**Used as a vector image tag.** Points to a Tektronix 4015 data. The bytes in the data field, when read and sent to a Tektronix 4015 terminal, will be displayed as a vector image.

TABLE D

The HDF Scientific Data Set Tags

Tag Name	Short Description	Data Size	Tag Value	Long Description
DFTAG_NDG	Numeric Data Group	n*4 bytes	720	Lists the Data Identifiers (tag/reference number pairs) that describe a scientific dataset. Supersedes DFTAG_SDG.
DFTAG_SDD	SDS Dimension Record	n bytes	701	Defines the rank and dimensions of the array the corresponding SD refers to.
DFTAG_SD	Scientific Data	Real Number	702	Points to scientific data.
DFTAG_SDS	SCales	Real Number	703	Identifies the scales to be used when interpreting and displaying data.
DFTAG_SDL	Labels	String	704	Labels all dimensions and data.
DFTAG_SDU	Units	String	705	Displays units for all dimensions and data.
DFTAG_SDF	Formats	String	706	Displays formats for axes and data.
DFTAG_SDM	Maximum/minimum	2 Real Numbers	707	Displays the maximum and minimum values for the data.
DFTAG_SDC	Coordinate system	String	708	Displays the coordinate system to be used in interpreting data.
DFTAG_SDLNK	SDS Link	8 bytes	710	Links and old-style DFTAG_SDG and a DFTAG_NDG in cases where the DFTAG_NDG meets all criteria for a DFTAG_SDG.
DFTAG_CAL	Calibration Information	36 bytes	731	The calibration record for the corresponding DFTAG.SD.
DFTAG_FV	Fill Value	n bytes	732	The value which has been used to indicate unset values in the corresponding DFTAG_SD.

TABLE E

The HDF Vset Tags

Tag Name	Short Description	Data Size	Tag Value	Long Description
DFTAG_VG	Vgroup	14+n bytes	1965	Provides a general-purpose grouping structure.
DFTAG_VH	Vdata Description	22+n bytes	1962	Provides information necessary to process a DFTAG_VS.
DFTAG_VS	Vdata	n bytes	1963	Contains a block a data that is to be interpreted according to the information in the corresponding DFTAG_VH.

TABLE F

The Obsolete HDF Tags

Tag Name	Short Description	Data Size	Tag Value	Long Description
DFTAG_IDS	Image Dimension-8	4 bytes	200	Two 16-bit integers that represent the width and height of an 8-bit raster image in bytes.
DFTAG_IP8	Image Palette-8	768 bytes	201	A 256 x 3 byte array representing the red, green and blue elements of the 256-color palette respectively.
DFTAG_RI8	Raster Image-8	x*y bytes	202	A row-oriented representation of the elementary 8-bit image data.

Tag Name	Short Description	Data Size	Tag Value	Long Description
DFTAG_CI8	Compressed Image-8	n bytes	203	A row-oriented representation of the elementary 8-bit raster image data, with each row compressed using a form of run-length encoding.
DFTAG_II8	IMCOMP Image-8	n bytes	204	A 4:1 8-bit raster image, compressed using the IMCOMP algorithm.
DFTAG_SDG	Scientific Data Group	n*4 bytes	700	List the Data Identifiers (tag/reference number pairs) that uniquely describe a scientific dataset.
DFTAG_SDT	Transpose	0 bytes	709	Indicates that data is transposed in the file.

Appendix B: HDF Installation Overview

General HDF Installation Overview

A. Acquiring the HDF Library Source.

You may obtain HDF via FTP, an archive server, or US mail.

FTP server: If you are connected to Internet (NSFNET, ARPANET, MILNET, etc.) you may download HDF source code at no charge from the anonymous ftp server at NCSA. The Internet address of the server is:

```
ftp.ncsa.uiuc.edu or 141.142.3.135
```

Note: the IP address has been changed from 141.142.20.50 to 141.142.3.135. Log in by entering anonymous for the name and your local e-mail address ("login@host") for the password.

After logging in change directory to "HDF/HDF4.1/". If you want packed source code, change directory to "tar/", "hqx/", or "zip/". Files in those directories must be transferred using binary mode.

If you want unpacked source code, change directory to "unpacked/" and transfer all the files in unpacked/ and in its subdirectories to your host.

If you have any questions regarding this procedure or whether you are connected to Internet, consult your local system administration or network expert.

We have set up an HDF anonymous FTP server to mirror what is on the FTP server. The internet address of the HDF server is:

```
hdf.ncsa.uiuc.edu (141.142.21.14)
```

If you try to log on to NCSA's ftp server and receive the message saying too many users are connected to NCSA ftp, try this HDF ftp server.

Archive server: E-mail a request to: archive-server@ncsa.uiuc.edu Include in the subject or message line the word "help", then press RETURN. Send another e-mail request to: `archive-server@ncsa.uiuc.edu` Include in the subject or message line the word "index", then press RETURN. The information you receive from both the help and index commands will give you further instructions on obtaining NCSA software. Refer to Chapter 1 of *HDF Calling Interfaces Manual* for details.

US mail: A tape or CDROM archive of HDF is also available for purchase through the NCSA technical Resources Catalog. To obtain a catalog, contact:

```
NCSA Documentation Orders
152 Computing Applications Building
605 East Springfield Avenue Champaign, IL 61820
(217) 244-4130
```

B. Building the HDF Library Source.

The HDF base library can be built with a single command from the top level directory where the subdirectories "src/", "util/", and "test/" reside. The file "Makefile.template" is a generic, machine independent makefile which you can modify if there is no makefile already built for your machine.

For convenience, there are also machine-customized makefiles. For example. the "MAKE.IBM6000" file is a makefile suitable for compiling HDF on an IBM RS/6000. Assuming you are on an IBM RS/6000, copy "MAKE.IBM6000" to "Makefile" and use the following commands to install different targets:

```
cp MAKE.IBM6000 Makefile
```

`make allnofortran` builds the HDF library and only the C interfaces, the utilities and the C test programs. make all builds the HDF library with the C and Fortran-77 interfaces, the utilities, and C and Fortran-77 test programs.

General netCDF Installation Overview

Building the netCDF Library Source.

The HDF netCDF/HDF library build process is automatically configured by the makefile system. Modify the file named "CUSTOMIZE" in the "mfhdf" (which stands for ***multi-file hdf***) directory and run the script named "configure". It will set up all of the makefiles correctly. Refer to the "INSTALL", "README", and "README.HDF" files in the "mfhdf" directory for instruction and direction.

Procedures to Set Up the HDF Application Programming Environment

To use HDF routines in your C program, you must add the line `#include "hdf.h"` if you don't use the netCDF/HDF library, or `#include "mfhdf.h"` otherwise. This must be near the beginning of your code.

Note: Applications that need netCDF or multi-file SDS functionality should link with both "libnetcdf.a" and "libdf.a" **in this order** (the order is critical!). Applications that use neither of these interfaces can just link with the "libdf.a" library for the base level of HDF functionality.

If you are on a SUN SPARC, the include files are in the directory "incdir", the base library file "libdf.a" is in "libdir", and the netCDF/HDF library file "libnetcdf.a" is in "mflibdir". Use the following command to compile a C program "myprog.c":

```
cc -DSUN -DHDF -Iincdir myprog.c mflibdir/libnetcdf.a libdir/libdf.a -o myprog
```

or

```
cc -DSUN -DHDF -Iincdir myprog.c -L mflibdir -lnetcdf -L libdir -ldf -o myprog
```

The `mflibdir/libnetcdf.a` or `-L mflibdir -lnetcdf` need not be included if you are not using the multi-file interface.

For Fortran-77 programs, if your Fortran-77 compiler accepts 'include' statements, you may include :hdf.inc", "dffunc.inc", and "netcdf.inc" in your program. Otherwise, you need to declare in your program all the constants used and functions called by the program. To compile a Fortran-77 program "myprogf.f" use:

```
f77 -o myprogf myprogf.f mflibdir/libnetcdf.a libdir/libdf.a
```

or

```
f77 -o myprogf myprogf.f -L mflibdir -lnetcdf -L libdir -ldf
```

Again, the `mflibdir/libnetcdf.a` need not be included if you are not using the multifile interface.

Windows NT Installation

Please refer to the "install_winNT.txt" file in the "./release_notes" directory of the distribution.

Use of the Pablo Instrumentation of HDF

This version of the distribution has support for creating an instrumented version of the HDF library (libdf-inst.a). This library, along the Pablo performance data capture libraries, can be used to gather data about I/O behavior and procedure execution times.

More detailed documentation on how to use the instrumented version of the HDF library with Pablo can be found in the Pablo directory '$(toplevel)/hdf/pablo'. See the provided '$(toplevel)/hdf/pablo/README.Pablo' and the Postscript file '$(toplevel)/hdf/pablo/Pablo.ps'.

At this time only an instrumented version of the core HDF library libdf.a can be created. Future versions will have support for the SD interface found in libmfhdf.a. Current interfaces supported are AN, GR, DFSD, DFAN, DFP, DFR8, DF24, H, V, and VS.

To enable the creation of an instrumented library the following section in the makefile fragment($(toplevel)/config/mh-<os>) must be uncommented and set.

```
# ------------ Macros for Pablo Instrumentation  --------------------
# Uncomment the following lines to create a Pablo Instrumentation
# version of the HDF core library called 'libdf-inst.a'
# See the documentation in the directory 'hdf/pablo' for further
# information about Pablo and what platforms it is supported on
# before enabling.
# You need to set 'PABLO_INCLUDE' to the Pablo distribution
# include directory to get the files 'IOTrace.h' and 'IOTrace_SD.h'.
#PABLO_FLAGS  = -DHAVE_PABLO
#PABLO_INCLUDE = -I/hdf2/Pablo/Instrument.HP/include
```

After setting these values you must re-run the top-level 'configure' script. Make sure that your start from a clean re-build(i.e. 'make clean') after re-running the toplevel 'configure' script and then run 'make'. Details on running configure can be found in the section 'General Configuration/Installation - Unix' found in the top-level installation file '$(toplevel)/INSTALL'.