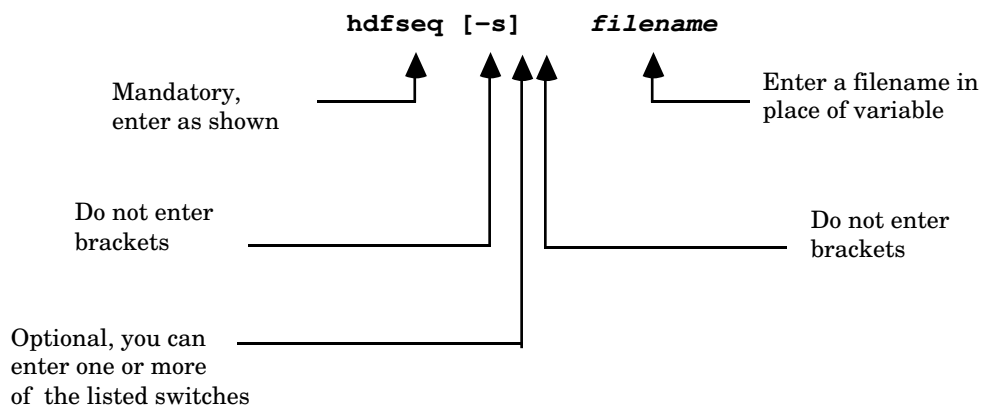


On separate line or within a text line; italic, lowercase, courier bold type	<i>filename</i>	This notation is a variable, which represents a certain kind of entry, but may consist of different characters every time you make the entry.
Square brackets; bold type	[]	Do not enter the square brackets. Material presented between brackets is optional and should be entered only in certain cases.
Ellipses; bold type	...	Do not enter ellipses. Ellipses indicate you may enter more material similar to the material preceding the ellipses.

Figure 1 shows you how to read and enter a command line.

Figure I.1 Reading and Entering a Command Line



Further Reading

More detailed information about the basic HDF structure and how it works can be found in *NCSA HDF Specifications*, which you may download via FTP (refer to "How to Get HDF" in Chapter 1) or may request by writing NCSA at the addresses listed on the Bugs and Suggestions report form at the back of this manual.

use with NCSA HDF.

Appendix F, “NCSA HDF README Files on Anonymous FTP,” includes listings of the README files which can be found in the anonymous FTP directory that contains NCSA HDF.

Form of Presentation

The material in this manual is presented in text, screen displays, or command line notation.

Text

In explaining various features and commands, this manual often presents a word within a paragraph in *italics* to indicate that the word is defined within the paragraph, or that it is a significant term that should be noted and/or is being mentioned for the first time.

So that they are more easily identifiable within this manuscript, utilities, variables, integer arguments, tags, etc. have been presented in 10 point courier style.

Portions of this manual refer to other portions of the manual where the other portions explain related topics. These cross references usually mention the title of sections or chapters enclosed in quotation marks, such as, See Chapter 1, “NCSA HDF Basics.”

Command Line Format Notation

Throughout this manual, many explanations instruct you to make entries by typing on the keyboard. These entry instructions are printed in **courier bold type** and appear within a paragraph or on a separate line. The command lines in this manual are normally shown in lowercase, except in rare instances where uppercase is required.

When it is necessary for you to press a key that is labeled with more than one character (such as the RETURN key), this manual identifies the key with all capital (uppercase) letters. Keys to be entered are printed in bold type. Keys that are pressed simultaneously or in succession are linked with a hyphen. For example, press CONTROL-A.

The meaning of each special notation applied to format lines is listed in Table 1.

Table I.1 Meaning of Entry Format Notations

Appearance	Example	Entry Method
On separate line; lowercase, courier bold type	dothis	Enter the keys for each character.
Within a text line; uppercase	RETURN	Press the single key indicated.
Within a text line; uppercase; hyphens between key names	CONTROL-A	While holding down the first one or two key(s) indicated, press the last key indicated.

Use of This Manual

This manual is designed for users who are working on an application that involves the use of HDF files. If you are completely unfamiliar with HDF, you should read or scan Chapter 1, “HDF Basics.”

The most important information about a particular interface is covered in the chapter that deals with that interface. Each chapter contains a description of the set with which the interface deals, descriptions of the routines that constitute the interface, and examples.

Manual Contents

The manual is organized into the following chapters:

Chapter 1, “NCSA HDF Basics,” describes HDF, explains the reasons for its creation, and gives a brief overview of available applications software.

Chapter 2, “Storing Raster Images,” describes the routines that are available for storing and retrieving raster images in HDF format, with or without compression, and with or without palettes.

Chapter 3, “Storing Palettes,” describes the routines that are available for storing and retrieving 8-bit palettes.

Chapter 4, “Storing Rectangular Gridded Arrays of Scientific Data,” describes the routines that are available for storing and retrieving scientific datasets.

Chapter 5, “Annotating Data Objects and Files,” describes the routines that are available for storing and retrieving data and file annotations.

Chapter 6, “General Purpose HDF Routines,” provides a reference to the set of general purpose low-level routines you’ll use when working with HDF files.

Chapter 7, “NCSA HDF Command Line Utilities,” lists currently available utility routines and command line utilities that are available for working with HDF files.

Chapter 8, “Error Handling in HDF,” discusses the error reporting procedures of HDF.

Appendix A, “NCSA HDF Tags,” presents a table of brief descriptions of most of the tags assigned at NCSA for general use.

Appendix B, “Header Files and FORTRAN Constants File,” includes the general header files used in compiling all HDF libraries.

Appendix C, “Eight-Character FORTRAN Names,” lists the official identifier names and their short form equivalents.

Appendix D, “Public HDF Directories on NCSA Computers,” lists the names and contents of NCSA-supported HDF directories.

Appendix E, “Routine Lists,” lists all the current routines available for

ing-point data to images, 24-bit images to 8-bit images, and raw palettes to and from HDF palettes

- Minor routines to help fill out existing interfaces

Differences Between HDF 3.1 and HDF 3.0

These are the changes made in release 3.1:

- Added function `DFR8readref`, `DF24readref`, `DFSDreadref`, and `DFPreadref` to permit random access by reference number
- Added the functions `DF24restart` and `DF24setil`
- Speeded up the `DFSDgetdata`, `DFSDputdata`, `DFSDadddata`, `DFSDgetslice` and `DFSDputslice` functions, especially for UNICOS machines
- Added several new annotation functions: `DFANaddfid`, `DFABaddfds`, `DFANgetfidlen`, `DFANgetfid`, `DFANgetdslen`, `DFANgetfds`, `DFANaddfann`, `DFANgetfannlen`, `DFANgetfann` and `DFANlastref`
- Revised `DFANlablist` so that it returns all ref numbers for a given tag, regardless of whether or not there is a label
- Fixed bug with `DFSDgetdata` so that it always goes to the next SDS
- Fixed bug concerning checking the status of opening a file with unbuffered i/o
- Added some macros to make passing character arrays from FORTRAN to C easier
- Fixed some more minor bugs
- Recoded some parts for cosmetic reasons.

System Requirements

In order to use HDF software, you need access to one of the following computer systems: a Cray with UNICOS, Silicon Graphics system with UNIX, an Alliant with CONCENTRIX, a Sun System 3 with UNIX, a VAX terminal with UNIX, a VAX with VMS, a Macintosh with MacOS, or an IBM PC with MS-DOS, or compatible model.

HDF has been ported to other platforms, but they are not officially supported by NCSA. A HINTS file on anonymous ftp contains information on porting HDF to other platforms. (See the section “How to Get HDF” for information on how to access NCSA’s anonymous ftp server.)

Introduction

Overview

This introduction provides an overview of NCSA Hierarchical Data Format (HDF) capabilities on the Cray, Alliant, Sun, Iris, Macintosh, and IBM PC. The organization and use of this manual are described and notational conventions explained.

What Is HDF?

Hierarchical Data Format (HDF) is a multi-object file format for the transfer of graphical and floating-point data between machines. The design of this format allows self-definition of data content, and easy extensibility for future enhancements or compatibility with other standard formats.

Features of NCSA HDF Version 3.0 include the following:

- Facilities for sharing data across machines and systems, including Cray (UNICOS), Silicon Graphics (UNIX), Alliant (CONCENTRIX), Sun (UNIX), VAX (UNIX), Macintosh (MacOS), and IBM PC (MS-DOS)
- FORTRAN and C calling interfaces for storing and retrieving 8- and 24-bit raster images, palettes, scientific data and accompanying annotations
- Utilities for editing and displaying HDF files, and converting raw data files to HDF files and vice versa
- General purpose routines for creating HDF files

Differences Between HDF 2.0 and HDF 3.0

All of the features of HDF 2.0 are present in HDF 3.0. HDF 3.0 contains the following new features:

- An interface for basic input and output of 24-bit raster images
- An interface for annotating HDF data objects and files
- An interface for input and output of 8-bit palettes
- Scientific dataset routines for storing and retrieving subsets (slices) of scientific data, and for choosing optional storage formats and data types
- New utilities, including an HDF editor, and routines to convert float-