## Manual Contents

The manual is organized into the following chapters:

Chapter 1, "The Basic Structure of HDF Files," introduces and describes the components and organization of Hierarchical Data Format files.

Chapter 2, "The NCSA HDF General Purpose Interface," contains a list of the routines that make up the low level interface described in detail in the HDF manual.

Chapter 3, "Sets and Groups," describes raster image sets and scientific datasets, explains the role of sets and groups in an HDF file, and discusses the routines that are available for storing and retrieving scientific datasets.

Chapter 4, "HDF Conventions," presents guidelines regarding the use of HDF that are not discussed elsewhere.

Chapter 5, "Other Formats," briefly describes the following formats and compares their features with those of HDF: CGM, PICT2, TIFF, CDF, FITS, PHIGS, and PHIGS+.

Appendix A, "NCSA HDF Tags," presents a list of brief descriptions of the tags assigned at NCSA for general use.

Appendix B, "Header Files," includes the general header files used in compiling all HDF libraries.

## Form of Presentation

The material in this manual is presented in text or screen displays.

### Text

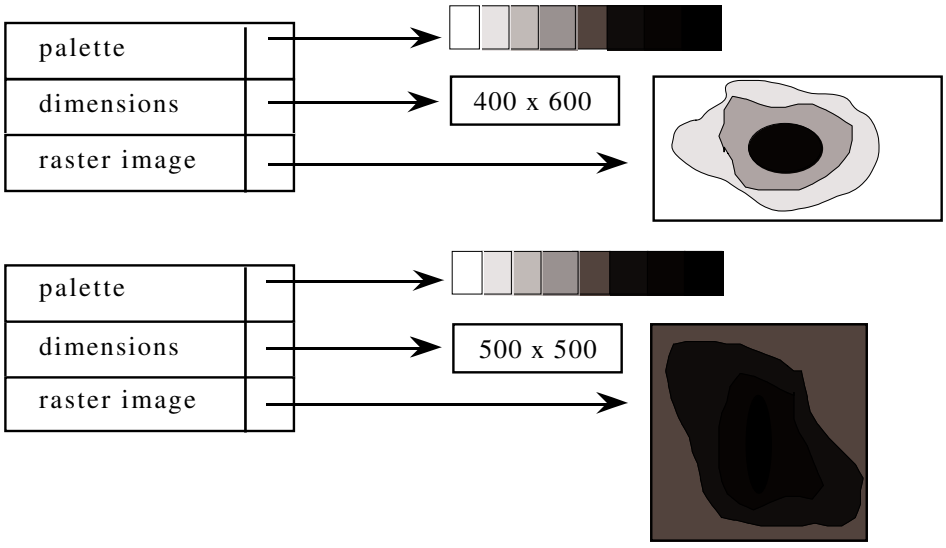
In explaining various features and commands, this manual often presents a word within a paragraph in *italics* to indicate that the word is defined within the paragraph.

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, "The Basic Structure of HDF Files."

### Screen Displays

Screen displays in this manual are presented in `courier type`.

**Figure 1** Raster Image Sets in an HDF File




HDF files never need to become out of date. For example, suppose a site falls far behind in the HDF standard, so its users can only work with the portions of the specification that are three years old. Users at this site might want to import files from NCSA. Even with the more advanced data files, they can list the types of data in the file. All of the older types of data objects that they understand are still accessible, despite the fact that the older types of data objects are mixed in with new kinds of data. In addition, if the more advanced site uses the text annotation facilities of HDF effectively, the files will arrive with complete human-readable descriptions of how to decipher the new tag types.

To present a convenient user interface made up of something other than a list of tag types with their associated data requirements, HDF supports multiple calling interfaces. The low level calling interface for manipulating tags and raw data is designed to be used by systems programmers who are providing the higher level interfaces for applications like raster image storage or scientific data archiving.

An important issue in data file design is that of machine independence or transportability. The HDF design is not machine independent, but it defines the data completely. HDF requires you to fully specify all number types used, so conversion programs can identify what number formats are being used and do the conversions when needed.

## Use of This Manual

This manual is designed for software developers who are designing applications or routines for use with HDF files and for users who need detailed information about HDF.

# Introduction

## Overview

At the National Center for Supercomputing Applications, diverse groups are working on supercomputing projects of many kinds. The Hierarchical Data Format (HDF) was designed to make the sharing of data between different people, different projects, and different types of computers easy and self-describing. An extensible header, along with carefully crafted internal layers, provides a system that can grow along with the software that NCSA develops. This chapter provides a brief overview of HDF capabilities and design.

## What Is HDF?

HDF is a growth-oriented approach to file format design. Rather than try to address all of the short-term issues in a fixed format, or attempt to solve all of the hard problems in an all-purpose format, HDF defines a framework for continued future growth. New calling interfaces can be defined that are compatible with old ones. Files can be made upwardly compatible for years to come without giving up added power in the future. HDF also makes it easy for the user to include annotations, titles, and specific descriptions of the data in the file, so that files can be archived with human-readable information about their origins.

HDF uses the concept of a tagged, or *object oriented*, file organization. The idea is to store both a known format description and the data in the same file. HDF tags describe the format of the data because each tag is assigned a specific meaning—one tag is assigned to “File Identifier,” another is assigned to “Raster Image,” and so on (see Figure 1). A program that has been written to understand a certain list of tag types can scan the file for those tag types and process the data. This program also can ignore any data that is beyond its scope.