

Table of Contents

Introduction to HDF	1
1.1 Chapter Overview	1
1.2 What is HDF?	1
1.3 Why Was HDF Created?	3
1.4 High-Level HDF APIs	3
1.4.1 User's Guide Code Examples	4
1.5 NCSA HDF Command-Line Utilities and Visualization Tools	4
1.6 Current HDF Platforms	4
 HDF Fundamentals	 7
2.1 Chapter Overview	7
2.2 The Hierarchy of HDF Interaction	7
2.3 Data Objects	8
2.3.1 Data Descriptors	8
2.4 File Format	9
2.4.1 Grouping Data Objects in an HDF File	9
2.4.2 Storing Data Objects	10
2.4.2.1 Contiguous Data Elements	10
2.4.2.2 Linked-block Data Elements	10
2.4.2.3 External Data Elements	10
2.5 Header File Information	11
2.5.1 File Access Code Definitions	11
2.5.2 Data Type Definitions	11
2.5.3 Tag Definitions	12
2.5.4 Limit Definitions	13
2.6 Basic Operations on HDF Files	13
2.6.1 File Identifiers	13
2.6.2 Opening HDF Files: Hopen	13
2.6.3 Closing HDF Files: Hclose	14
2.6.4 Determining the Number of Objects with a Specified Tag: Hnumber	14
2.6.5 Getting the HDF Library Version Used to Create a File: Hgetlibversion	14
2.6.6 Locate an Object by its Tag/Reference Number Pair: Hfind	15
2.7 Application Programming Interfaces	15
2.8 Fortran-77 and C Language Issues	15
2.8.1 Fortran-77-to-C Translation	15
2.8.2 Case Sensitivity	16
2.8.3 Name Length	16
2.8.4 Header Files	16
2.8.5 Data Type Specifications	16
2.8.6 Array Specifications	17
2.8.7 Fortran-77, ANSI C and K&R C	17
2.9 Low-Level Interfaces	17
2.9.1 The H Interface	17
2.9.2 The HL/HX Interface	18
2.9.3 The HE Interface	18
 Scientific Data Sets (SD API)	 19
3.1 Chapter Overview	19

3.2	The SD Scientific Data Set Data Model	19
3.2.1	Required SD SDS Objects	19
3.2.1.1	SDS Array	20
3.2.1.2	SDS Array Name	20
3.2.1.3	Data Type	20
3.2.1.4	Dimensions	20
3.2.2	Optional SD SDS Objects	20
3.2.3	Annotations and the SD Data Model	21
3.3	The SD Scientific Data Set API	21
3.3.1	SD Library Routines	21
3.3.2	SDS Identifiers in the SD Interface	23
3.3.3	Tags in the SD Interface	23
3.4	Programming Model for the SD Interface	23
3.4.1	Establishing Access to Files and Data Sets: SDstart and SDselect	24
3.4.2	Terminating Access to Files and Data Sets: SDendaccess and SDend	25
3.5	Creating and Writing to Simple Scientific Data Sets	26
3.5.1	Creating Scientific Data Sets: SDcreate	26
3.5.2	Writing Data to an SDS Array: SDwritedata	28
3.5.2.1	Filling an Entire Array	30
3.5.2.2	Writing Slabs to an SDS Array	32
3.5.2.3	Appending Data to an SDS Array Using the Unlimited Dimension	36
3.5.3	Compressing SD SDS Array Data: SDsetcompress	40
3.5.3.1	Rules for Writing to a Compressed Data Set	41
3.5.4	External File Operations	41
3.5.4.1	Specifying the Directory Search Path of an External File: HXsetdir	42
3.5.4.2	Specifying the Location of the Next External File to be Created: HXsetcreatedir	42
3.5.4.3	Creating a Data Set in an External File: SDsetexternalfile	43
3.5.4.4	Moving Data to an External File	44
3.6	Reading Data from an SDS Array: SDreaddata	45
3.6.1	Reading Data from an External File	51
3.7	Obtaining Information About SD Data Sets	51
3.7.1	Obtaining Information About the SDSs in a File: SDfileinfo	51
3.7.2	Obtaining Information About a Specific SDS: SDgetinfo	52
3.7.3	Locating a SDS Data Set by Name: SDnametoindex	54
3.7.4	Locating an SDS by Reference Number: SDreftoindex	54
3.7.5	Creating SDS Arrays Containing Variable-Length Data: SDsetnbitdataset	56
3.8	Chunked (or Tiled) Scientific Data Sets	57
3.8.1	Making a Non-Chunked SDS a Chunked SDS: SDsetchunk	58
3.8.2	Setting the Maximum Number of Chunks in the Cache: SDsetchunkcache	59
3.8.3	Writing Data to Chunked SDSs: SDwritechunk and SDwritedata	59
3.8.4	Reading Data From Chunked SDSs: SDreadchunk and SDreaddata	60
3.8.5	Obtaining Information About a Chunked SDS: SDgetchunkinfo	61
3.8.6	Ghost Areas	69
3.9	SD Dimension and Dimension Scale Operations	70
3.9.1	Selecting a Dimension: SDgetdimid	70
3.9.2	Naming a Dimension: SDsetdimname	70
3.9.3	Old and New Dimension Implementations	71
3.9.3.1	Setting the Future Compatibility Mode of a Dimension: SDsetdimval_comp	71
3.9.3.2	Setting the Current Compatibility Mode of a Dimension: SDisdimval_bwcomp	72
3.9.4	Dimension Scales	72
3.9.4.1	Writing Dimension Scales: SDsetdimscale	72
3.9.4.2	Obtaining Dimension Scale and Other Dimension Information: SDdiminfo	73

3.9.4.3	Reading Dimension Scales: SDgetdimscale	73
3.9.5	Distinguishing SDS Arrays from Dimension Scales: SDiscoordvar	76
3.9.6	Dimension Scales for Multiple Data Sets	78
3.10	User-defined Attributes	79
3.10.1	Writing User-defined Attributes: SDsetattr	80
3.10.2	Querying User-defined Attributes: SDfindattr and SDattrinfo	82
3.10.3	Reading User-defined Attributes: SDreadattr	82
3.11	Predefined Attributes	85
3.11.1	Accessing Predefined Attributes	86
3.11.2	SDS String Attributes	87
3.11.2.1	Writing String Attributes: SDsetdatastrs	87
3.11.2.2	Reading String Attributes: SDgetdatastrs	87
3.11.3	Dimension String Attributes	88
3.11.3.1	Writing a Dimension String Attribute: SDsetdimstrs	88
3.11.3.2	Reading a Dimension String Attribute: SDgetdimstrs	88
3.11.4	Range Attributes	89
3.11.4.1	Writing a Range Attribute: SDsetrange	89
3.11.4.2	Reading a Range Attribute: SDgetrange	89
3.11.5	Fill Values	89
3.11.5.1	Writing a Fill Value Attribute: SDsetfillvalue	90
3.11.5.2	Reading a Fill Value Attribute: SDgetfillvalue	90
3.11.5.3	Setting the Fill Mode for all SDSs in the Specified File: SDsetfillmode	90
3.11.6	Calibration Attributes	91
3.11.6.1	Writing Calibrated Data: SDsetcal	91
3.11.6.2	Reading Calibrated Data: SDgetcal	92
3.12	netCDF	93
3.12.1	HDF Interface vs. netCDF Interface	94

Vdatas (VS API) 97

4.1	Chapter Overview	97
4.2	The Vdata Model	97
4.2.1	Records and Fields	98
4.2.2	Conventions	98
4.3	The Vdata API	98
4.3.1	Vdata Library Routines	99
4.3.2	Vdata Identifiers in the Vdata API	100
4.3.3	Programming Model for the Vdata Interface	100
4.3.4	Accessing Files and Vdatas: Hopen, Vstart and VSattach	101
4.3.5	Terminating Access to Vdatas and Files: VSdetach, Vend and Hclose	101
4.4	Creating and Writing to Single-Field Vdatas: VHstoredata and VHstoredatam	103
4.5	Creating and Writing to Multi-Field Vdatas	107
4.5.1	Creating Vdatas	107
4.5.1.1	Creating New Vdatas with VSattach	108
4.5.1.2	Assigning a Vdata Name and Class: VSsetname and VSsetclass	108
4.5.1.3	Defining a Field Within Vdatas: VSfdefine	108
4.5.1.4	Initializing the Fields for Write Access: VSsetfields	108
4.5.1.5	Specifying the Interlace Mode: VSsetinterlace	109
4.5.2	Writing Data to Vdatas	110
4.5.2.1	Resetting the Current Position Within Vdatas: VSseek	110
4.5.2.2	Writing to a Vdata: VSwrite	111

4.5.2.3	Filling Records of Mixed Field Data Types	115
4.6	Reading from Vdatas	118
4.6.1	Selecting the Record to Begin Reading from: VSseek	119
4.6.2	Selecting the Fields to be Read: VSsetfields	119
4.6.3	Reading from the Current Vdata: VSread	119
4.7	Searching for Vdatas in a File	122
4.7.1	Finding All Vdatas that are Not Members of Vgroups: VSlone	122
4.7.2	Searching for the Reference Number of a Vdata: VSgetid	123
4.7.3	Determining a Reference Number from a Vdata Name: VSfind	123
4.7.4	Searching for a Vdata by Field Name: VSfexist	123
4.8	Vdata Attributes	125
4.8.1	Querying the Index of a Vdata Field Given the Field Name: VSfindx	125
4.8.2	Setting the Attribute of a Vdata Field or Vdata: VSsetattr	126
4.8.3	Querying the Total Number of Vdata Attributes: VSnattrs	126
4.8.4	Querying the Number of Attributes of a Vdata or a Vdata Field: VSfnattrs	127
4.8.5	Retrieving the Index of a Vdata Attribute Given the Attribute Name: VSfindattr	127
4.8.6	Querying Information on a Given Vdata Attribute: VSattrinfo	128
4.8.7	Querying the Values of a Given Vdata Attribute: VSgetattr	128
4.8.8	Determining if the Given Vdata is an Attribute: VSiattr	129
4.9	Obtaining Information About a Specific Vdata	131
4.9.1	Obtaining Vdata Information: VSinquire	132
4.9.2	VSQuery Vdata Information Retrieval Routines	134
4.9.3	VS Vdata Information Retrieval Routines	135
4.9.4	VF Field Information Retrieval Routines	136

Vgroups (V API) 137

5.1	Chapter Overview	137
5.2	The Vgroup Data Model	137
5.2.1	Vgroup Names and Classes	137
5.2.2	Vgroup Organization	138
5.2.3	Vgroup Conventions: An Example Using Vsets	138
1.3	The Vgroup API	139
1.3.1	Vgroup Library Routines	139
1.3.2	Vgroup Identifiers in the Vgroup Interface	140
1.4	Programming Model for the Vgroup Interface	141
1.4.1	Accessing Files and Vgroups: Hopen, Vstart and Vattach	141
1.4.2	Terminating Access to Vgroups and Files: Vdetach, Vend and Hclose	142
1.5	Creating and Writing to a Vgroup	142
1.5.1	Creating a Vgroup: Vattach	142
1.5.2	Assigning a Vgroup Name and Class: Vsetname and Vsetclass	143
1.5.3	Inserting Any HDF Data Object Into a Vgroup: Vaddtagref	143
1.5.4	Inserting a Vdata or Vgroup Into a Vgroup: Vinsert	144
1.6	Reading from Vgroups	151
1.6.1	Locating Vgroups in Files	151
1.6.1.1	Locating a Lone Vgroup: V lone	151
1.6.1.2	Determining the Next Vgroup Identifier: Vgetid	151
1.6.1.3	Determining a Vgroup's Name and Class: Vgetname and Vgetclass	151
1.7	Vgroup Attributes	154
1.7.1	Querying Information on a Given Vgroup Attribute: Vattrinfo	154
1.7.2	Querying the Vgroup Version of a Given Vgroup: Vgetversion	154

1.7.3	Querying the Total Number of Vgroup Attributes: Vnattrs	155
1.7.4	Querying the Values of a Given Vgroup Attribute: Vgetattr	155
1.7.5	Setting the Attribute of a Vgroup: Vsetattr	155
1.7.6	Retrieving the Index of a Vgroup Attribute Given the Attribute Name: Vfindattr	156
1.8	Obtaining Information About the Contents of a Vgroup	158
1.8.1	Returning the Tag/Reference Number Pairs of Vgroup Members: Vntagrefs, Vgettagref and Vgettagrefs	158
1.8.2	Returning Vgroup Member Information: Vinqtagref, Visvg and Visvs	159
1.9	The Vgroup Command-Line Utilities	161
1.9.1	Vshow	161
1.9.2	Vmake	161
1.10	Obsolete Vgroup Interface Routines	162
1.10.1	Determining the Next Vgroup or Vdata Identifier: Vgetnext	162
1.10.2	Determining the Number of Members and Vgroup Name: Vinquire	162
1.11	Vgroup Backward Compatibility Issues	163
1.11.1	Vset Implementation Integrated into the Vgroup Interface	163

8-Bit Raster Images (DFR8 API) 165

6.1	Chapter Overview	165
6.2	The 8-Bit Raster Data Model	165
6.2.1	Required 8-Bit Raster Image Data Set Objects	165
6.2.1.1	8-Bit Raster Image Data Representation	165
6.2.1.2	8-Bit Raster Image Dimension	166
6.2.2	Optional 8-Bit Raster Image Data Set Objects	166
6.2.2.1	Palettes	166
6.2.3	Compression Method	167
6.3	The 8-Bit Raster Image Interface	168
6.3.1	8-Bit Raster Image Library Routines	168
6.4	Writing 8-Bit Raster Images	168
6.4.1	Storing a Raster Image: DFR8putimage and DFR8addimage	169
6.4.2	Adding a Palette to an RIS8 Object: DFR8setpalette	170
6.4.3	Compressing 8-Bit Raster Image Data: DFR8setcompress	172
6.4.4	Specifying the Reference Number of an RIS8: DFR8writeref	175
6.5	Reading 8-Bit Raster Images	176
6.5.1	Reading a Raster Image: DFR8getimage	176
6.5.2	Querying the Dimensions of an 8-Bit Raster Image: DFR8getdims	176
6.5.3	Reading an Image with a Given Reference Number: DFR8readref	178
6.5.4	Specifying the Next 8-Bit Raster Image to be Read: DFR8restart	179
6.6	8-Bit Raster Image Information Retrieval Routines	179
6.6.1	Querying the Total Number of 8-Bit Raster Images: DFR8nimages	179
6.6.2	Determining the Reference Number of the Most-Recently-Accessed 8-Bit Raster Image: DFR8lastref	179
6.6.3	Determining the Reference Number of the Palette of the Most-Recently-Accessed 8-Bit Raster Image: DFR8getpalref	179
6.7	RIS8 Backward Compatibility Issues	180
6.7.1	Attribute "long_name" Included in HDF for netCDF Compatibility	180
6.7.2	Raster Image Group Implementation with New RIS8 Tags	180

24-Bit Raster Images (DF24 API) 181

7.1	Chapter Overview	181
7.2	The 24-Bit Raster Data Model	181
7.2.1	Required 24-Bit Raster Image Data Set Objects	181
7.2.1.1	24-Bit Raster Image Data Representation	181
7.2.1.2	24-Bit Raster Image Dimension	182
7.2.2	Optional 24-Bit Raster Image Data Set Objects	182
7.2.2.1	Compression Method	182
7.2.2.2	Interlace Modes	183
7.3	The 24-Bit Raster Interface	184
7.3.1	24-Bit Raster Image Library Routines	184
7.4	Writing 24-Bit Raster Images	184
7.4.1	Writing a 24-Bit Raster Image: DF24putimage and DF24addimage	185
7.4.2	Setting the Interlace Format: DF24setil	186
7.4.3	Compressing Image Data: DF24setcompress and d2sjpeg	188
7.5	Reading 24-Bit Raster Images	190
7.5.1	Reading a Raster Image: DF24getimage	190
7.5.2	Determining the Dimensions of an Image: DF24getdims	190
7.5.3	Modifying the Interlacing of an Image: DF24reqil	190
7.5.4	Reading a 24-Bit Raster Image with a Given Reference Number: DF24readref	192
7.5.5	Specifying that the Next Image Read to be the First 24-Bit Raster Image in the File: DF24restart	193
7.6	24-Bit Raster Image Information Retrieval Routines	193
7.6.1	Querying the Total Number of Images in a File: DF24nimages	193
7.6.2	Querying the Reference Number of the Most Recently Read or Written 24-Bit Raster Image: DF24lastref	193

General Raster Images (GR API) 195

8.1	Chapter Overview	195
8.2	The General Raster Data Model	195
8.2.1	Required General Raster Objects	195
8.2.1.1	Image Array	196
8.2.1.2	Image Array Name	196
8.2.1.3	Pixel Type	196
8.2.1.4	Dimensions	196
8.2.2	Optional General Raster Objects	196
8.3	The General Raster API	197
8.3.1	GR API Routines	197
8.4	Programming Model for the GR Interface	198
8.4.1	Accessing Files and Images: GRstart and GRselect	199
8.4.2	Terminating Access to Files and Images: GRenderaccess and GRender	199
8.5	Creating and Writing General Raster Images	200
8.5.1	Creating General Raster Images: GRcreate	200
8.5.2	Setting the Interlace Mode	201
8.5.3	Writing General Raster Images: GRwriteimage	203
8.6	Reading Data from an Image: GRreadimage	206
8.6.1	Reading General Raster Images from an External File	207
8.6.2	Setting the Interlace Mode for a Palette or Image Read	207
8.7	Obtaining Information About Files and General Raster Images	213

8.7.1	Obtaining Information about the Contents of a File: GRfileinfo	213
8.7.2	Obtaining Information About an Image: GRgetiminfo	214
8.7.3	Getting the Index of an Image: GRreftoindex and GRnametoindex	217
8.7.4	Compressing General Raster Images	220
8.7.5	External File Operations Using the GR API	220
8.7.5.1	Creating a General Raster Image in an External File	220
8.7.5.2	Moving General Raster Images to an External File	221
8.7.5.3	Setting the I/O Mode for External General Raster Image Access	221
8.8	General Raster Image Attributes	222
8.8.1	Predefined GR Attributes	222
8.8.2	Setting User-Defined Attributes: GRsetattr	222
8.8.3	Querying User-Defined Attributes: GRfindattr and GRattrinfo	224
8.8.4	Reading User-Defined Attributes: GRgetattr	224
8.9	Reading and Writing Palette Data Using the GR API	227
8.9.1	Obtaining a Palette ID: GRgetlutid	227
8.9.2	Obtaining Palette Information: GRgetlutinfo	227
8.9.3	Retrieving the Reference Number of the Specified Palette: GRluttoref	228
8.9.4	Writing Palette Data: GRwritelut	228
8.9.5	Reading Palette Data: GRreadlut	228
8.10	GR Object Identifier Conversion Functions	231
8.10.1	Mapping an Image Identifier to a Reference Number: GRidtoref	232
8.10.2	Mapping a Reference Number to an Index: GRreftoindex	232

Palettes (DFP API) 233

9.1	Chapter Overview	233
9.2	The Palette Data Model	233
9.3	The Palette API	234
9.3.1	Palette Library Routines	234
9.4	Writing Palettes	234
9.4.1	Writing a Palette: DFPaddpal and DFPputpal	234
9.4.2	Specifying the Reference Number of a Palette: DFPwriteref	236
9.5	Reading a Palette	236
9.5.1	Reading a Palette: DFPgetpal	236
9.5.2	Reading a Palette with a Given Reference Number: DFPreadref	237
9.5.3	Specifying the Next Palette to be Accessed to be the First Palette: DFPrestart	238
9.6	Other Palette Routines	238
9.6.1	Querying the Number of Palettes in a File: DFPnpals	238
9.6.2	Obtaining the Reference Number of the Most Recently Accessed Palette: DFPlastref	238
9.7	Backward Compatibility Issues	239

Annotations (DFAN and AN API) 241

10.1	Chapter Overview	241
10.2	The Annotation Data Model	241
10.2.1	Labels and Descriptions	241
10.2.2	File Annotations	241
10.2.3	Object Annotations	242
10.3	The Single-File Annotation Interface	242

10.3.1	Single-File Annotation Library Routines	243
10.3.2	Tags in the Annotation Interface	243
10.4	Programming Model for the Annotation Interface	244
10.5	Writing Annotations	244
10.5.1	Assigning a File Label: DFANaddfid	245
10.5.2	Assigning a File Description: DFANaddfds	245
10.5.3	Assigning an Object Label: DFANputlabel	246
10.5.4	Assigning an Object Description: DFANputdesc	246
10.6	Reading Annotations	249
10.6.1	Reading a File Label: DFANgetfidlen and DFANgetfid	249
10.6.2	Reading a File Description: DFANgetfdslen and DFANgetfds	249
10.6.3	Reading an Object Label: DFANgetlablen and DFANgetlabel	251
10.6.4	Reading an Object Description: DFANgetdesclen and DFANgetdesc	252
10.7	Maintenance Routines	254
10.7.1	Clearing the DFAN Interface's Internal Structures and Settings: DFANclear	254
10.8	Determining Reference Numbers	254
10.8.1	Determine a Reference Number for the Last Object Accessed: DF*lastref	255
10.8.2	Querying a List of Reference Numbers for a Given Tag: DFANlablist	256
10.8.3	Locate an Object by Its Tag and Reference Number: Hfind	258
10.9	The Multifile Annotation Interface	258
10.9.1	AN Library Routines	258
10.9.2	Type Definitions Used in the Multifile Annotation Interface	259
10.10	Programming Model for the Multifile Annotation Interface	259
10.11	Creating and Writing Annotations Using the AN Interface	259
10.11.1	Creating Annotations	260
10.11.2	Writing an Annotation: ANwriteann	261
10.12	Reading Annotations Using the AN Interface	263
10.12.1	Reading an Annotation: ANreadann	263
10.13	Obtaining Annotation Information Using the AN Interface	264
10.13.1	Selecting an Annotation: ANselect	264
10.13.2	Obtaining General Annotation Information: ANfileinfo	264
10.13.3	Getting the Length of an Annotation: ANannlen	265
10.13.4	Obtaining the Number of Annotations Corresponding to Given Search Criteria: ANnumann	267
10.13.5	Obtaining the List of Annotations Corresponding to Given Search Criteria: ANannlist	267
10.13.6	Obtaining the Tag/Reference Number Pair From a Specified Annotation Identifiers: ANget_tagref	269
10.13.7	Obtaining the Tag/Reference Number Pair From a Specified Annotation Identifier: ANid2tagref	269
10.13.8	Obtaining the Annotation Identifier From a Specified Tag/Reference Number Pair: ANtagref2id	270
10.13.9	Obtaining an Object Tag From a Specified Annotation Type: ANatype2tag	270
10.13.10	Obtaining an Annotation Type From a Specified Object Tag: ANtag2atype	270

Scientific Data Sets (DFSD API) 273

11.1	Chapter Overview	273
11.2	The DFSD Scientific Data Set Data Model	273
11.2.1	Required DFSD SDS Objects	273
11.2.1.1	Dimensions	274
11.2.2	Optional DFSD SDS Objects	274
11.2.2.1	Dimension Scales	274
11.2.2.2	Predefined Attributes	274
11.3	The Single-File Scientific Data Set Interface	274

11.3.1	DFSD Library Routines	274
11.3.2	File Identifiers in the DFSD Interface	275
11.4	Writing DFSD Scientific Data Sets	275
11.4.1	Creating a DFSD Scientific Data Set: DFSDadddata and DFSDputdata	276
11.4.2	Specifying the Data Type of a DFSD SDS: DFSDsetNT	276
11.4.3	Overwriting Data for a Given Reference Number: DFSDwriteref	278
11.4.4	Writing Several Data Sets: DFSDsetdims and DFSDclear	278
11.4.5	Preventing the Reassignment of DFSD Data Set Attributes: DFSDsetdims	279
11.4.6	Resetting the Default DFSD Interface Settings: DFSDclear	279
11.5	Reading DFSD Scientific Data Sets	280
11.5.1	Reading a DFSD SDS: DFSDgetdata	280
11.5.2	Specifying the Dimensions and Data Type of an SDS: DFSDgetdims and DFSDgetNT	280
11.5.3	Determining the Number of DFSD Data Sets: DFSDndatasets and DFSDrestart	282
11.5.4	Obtaining Reference Numbers of DFSD Data Sets: DFSDreadref and DFSDlastref	282
11.6	Slabs in the DFSD Interface	283
11.6.1	Accessing Slabs: DFSDstartslab and DFSDendslab	283
11.6.2	Writing Slabs: DFSDwriteslab	284
11.6.3	Reading Slabs: DFSDreadslab	285
11.7	Predefined Attributes and the DFSD Interface	286
11.7.1	Writing Data Set Attributes	286
11.7.1.1	Assigning String Attributes to an SDS: DFSDsetlengths and DFSDsetdatastrs	286
11.7.1.2	Assigning Value Attributes to a DFSD SDS: DFSDsetfillvalue, DFSDsetrange, DFSDsetcal	287
11.7.2	Reading DFSD Data Set Attributes	290
11.7.2.1	Reading Data Set Attributes: DFSDgetdatalen and DFSDgetdatastrs	290
11.7.2.2	Reading the Value Attributes of a DFSD Data Set: DFSDgetfillvalue and DFSDgetcal	292
11.7.3	Writing the Dimension Attributes of a DFSD SDS	293
11.7.3.1	Writing the String Attributes of a Dimension: DFSDsetlengths and DFSDsetdimstrs	293
11.7.3.2	Writing a Dimension Scale of a DFSD SDS: DFSDsetdimscale	294
11.7.4	Reading the Dimension Attributes of a DFSD SDS	294

Error Reporting 297

12.1	Chapter Overview	297
12.2	The HDF Error Reporting API	297
12.3	Error Reporting in HDF	297
12.3.1	Writing Errors to a File: HEprint	298
12.3.2	Returning the Code of the Nth Most Recent Error: HEvalue	298
12.3.3	Returning the Description of an Error Code: HEstring	299

HDF Performance Issues 305

13.1	Chapter Overview and Introduction	305
13.2	Examples of HDF Performance Enhancement	305
13.2.1	One Large SDS Versus Several Smaller SDSs	306
13.2.2	Sharing Dimensions Between Scientific Data Sets	307
13.2.3	Setting the Fill Mode	309
13.2.4	Disabling "Fake" Dimension Scale Values in Large One-Dimensional Scientific Data Sets	310
13.3	Data Chunking	312

13.3.1	What is Data Chunking?	312
13.3.2	Writing Concerns and Reading Concerns in Chunking	313
13.3.3	Chunking without Compression	313
13.3.3.1	Subsetting	313
13.3.4	Chunking with Compression	316
13.3.4.1	Use SDwritetechunk Rather Than SDwritedata to Write Compressed and Chunked SDSs	317
13.3.5	Chunk Sizing	317
13.3.5.1	Effect of Chunk Size on Performance	317
13.3.6	Chunk Cache Sizing	318
13.3.6.1	How Insufficient Chunk Cache Space can Impair Chunking Performance	318

HDF Command-Line Utilities 323

14.1	Chapter Overview	323
14.2	The HDF Command-Line Utilities	323
14.2.1	Listing Basic Information About an HDF File: hdflls	324
14.2.1.1	General Description	324
14.2.1.2	Command-Line Syntax	324
14.2.1.3	Examples	324
14.2.2	Editing the Contents of an HDF File: hdfed	325
14.2.2.1	General Description	325
14.2.2.2	Command-Line Syntax	326
14.2.3	Converting Floating-Point Data to SDS or RIS8: fp2hdf	334
14.2.3.1	General Description	334
14.2.3.2	Command-Line Syntax	335
14.2.3.3	Examples	336
14.2.4	Converting Several RIS8 Images to One 3D SDS: ristosds	336
14.2.4.1	General Description	336
14.2.4.2	Command-Line Syntax	336
14.2.4.3	Examples	337
14.2.5	Converting 8-Bit Raster Images to the HDF Format: r8tohdf	337
14.2.5.1	General Description	337
14.2.5.2	Command-Line Syntax	337
14.2.5.3	Examples	337
14.2.6	Extracting 8-Bit Raster Images and Palettes from HDF Files: hdftr8	338
14.2.6.1	General Description	338
14.2.6.2	Command-Line Syntax	338
14.2.6.3	Examples	339
14.2.7	Compressing RIS8 Images in an HDF File: hdfcomp	339
14.2.7.1	General Description	339
14.2.7.2	Command-Line Syntax	339
14.2.7.3	Examples	339
14.2.8	Converting 24-Bit Raw Raster Images to RIS8 Images: r24hdf8	339
14.2.8.1	General Description	339
14.2.8.2	Command-Line Syntax	339
14.2.8.3	Examples	340
14.2.9	Converting an HDF RIS24 Image to an HDF RIS8 Image: hdf24hdf8	340
14.2.9.1	General Description	340
14.2.9.2	Command-Line Syntax	340
14.2.10	Converting Raw Palette Data to the HDF Palette Format: paltohdf	340

14.2.10.1	General Description	340
14.2.10.2	Command-Line Syntax	340
14.2.11	Extracting Palette Data from an HDF File: hdfcpal	340
14.2.11.1	General Description	340
14.2.11.2	Command-Line Syntax	340
14.2.12	Compressing an HDF File: hdfpack	341
14.2.12.1	General Description	341
14.2.12.2	Command-Line Syntax	341
14.2.12.3	Examples	341
14.2.13	Displaying Vdata Information: vshow	341
14.2.13.1	General Description	341
14.2.13.2	Command-Line Syntax	341
14.2.13.3	Examples	342
14.2.14	Displaying General Information About the Contents of an HDF File: hdp	342
14.2.14.1	General Description	342
14.2.14.2	Command-Line Syntax	342
14.2.15	The HDF User-Contributed Utilities	344

Appendices 347

Appendix A: NCSA HDF Tags	347
Appendix B: HDF Installation Overview	352

