11. The ability to rename a single-file CDF with a different (or no) extension has been added. The CDF library will attempt to open a CDF by using the specification exactly as given as well as by appending `.cdf` automatically (see Section 2.2.5).

12. An operation has been added to the Internal Interface that allows the indexing statistics of a variable in a single-file CDF to be inquired. Section 2.2.6 describes the indexing scheme used for variables in a single-file CDF.

13. New EPOCH manipulation routines have been added that allow EPOCH character strings to be parsed and EPOCH values to be encoded. These routines are described in Sections 4.9 and 5.10 for C and Fortran applications, respectively.

14. The default qualifiers for the toolkit programs can now be configured when the CDF distribution is built and installed. These defaults can be overridden when a toolkit program is executed by specifying the desired qualifiers on the command line. The `README` file for each particular type of operating systems (VMS, UNIX, or MS-DOS) explains how to configure the CDF distribution.

15. An option has been added to the CDF library that allows the floating-point value `-0.0` to be converted to `0.0` automatically when detected. A value of `-0.0` is illegal on VAXes and DEC Alphas running OpenVMS. This option can be requested by an application (via the Internal Interface) or with a qualifier when running a toolkit program.

16. `CDFlist` has been modified to allow variable values to be displayed in either a vertical or horizontal format (when in zMode). The efficiency of `CDFlist` has also been improved (by using the `MONOTON` attribute or by the user specifying the monotonicity of the variables).

17. A qualifier has been added to the `CDFlist` and `CDFwalk` toolkit programs that allows the specification of the initial settings on the various menus available when running these programs.

18. A new qualifier has been added to the `CDFskeleton` toolkit program that allows `FILLVAL` attribute entry values to be used to specify the pad values for the corresponding variables.

19. A qualifier has been added to the `CDFstats` toolkit program that requests that the corresponding `MONOTON` attribute entries be updated based on the monotonicities detected for the variables.

20. A qualifier has been added to the `CDFcompare` toolkit program that requests that the locations (record/indices) of differences detected between two variables be displayed.

21. A qualifier has been added to the `CDFinquire` toolkit program that allows the specification of the variables for which values should be displayed.

22. The capability to use the `FORMAT` attribute has been added to all of the toolkit programs.

23. A qualifier has been added to most of the toolkit programs that allows a user to select which types of status codes from the CDF library should be reported: errors, warnings, and/or informationals.

24. The `CDFbrowse` toolkit program has been removed from the distribution. Its functionality remains by running `CDFedit` in a browsing mode.[1]

[1] The alias/symbol `CDFbrowse` still exists in the "definitions" file on UNIX/VMS systems but now executes `CDFedit` in a browse-only mode.

Appendix G

Glossary

assumed scope	An attribute scope used by the CDF library when it cannot determine the definite scope. This can occur when Version 1 CDFs are accessed.
attribute	A CDF object with which entries of metadata are associated.
big-endian	The byte ordering in which the most significant byte (MSB) is stored in the lowest memory location.
caching	The method used by the CDF library to improve performance when accessing a file. An attempt is made to keep commonly accessed blocks of the file in memory rather than repeatedly reading them from disk.
CDF	This term is used in more than one way... 1. The actual files that contain your data/metadata. For example: The CDF library must be used to create a "CDF." 2. The software distribution containing the CDF library, include files, and toolkit. For example: We like using "CDF" to store our data.
CDF distribution	The directory of software consisting of the CDF library, include files, and toolkit.
CDF library	The software library that is used to access a CDF.
CDF toolkit	A set of utility programs which ease the creation, modification, and verification of CDFs.
`CDF_OK`	A completion status code indicating unqualified success.
`cdf.h`	An include file used in C applications.
`cdf.inc`	An include file used in Fortran applications.

`cdfmsf.inc`	An include file used in Microsoft Fortran applications.
column-major	The variable majority where the first index of a multidimensional array of values increments the fastest.
conceptual view	The way that values along a dimension having a variance of `NOVARY` are made to appear as if they do actually exist (only one value is actually physically stored). This also applies to records beyond the last record actually stored. The conceptual view of a variable consists of "virtual" records and values (in addition to the physical records and values actually stored).
current	When the Internal Interface is used, current objects/states are those items affected when an operation is performed. For example, a current CDF is selected and then any operation performed involving a CDF is performed on that CDF (until a different current CDF is selected).
data specification	For a variable or attribute entry the data type and number of elements of that data.
data type	For a variable or attribute entry, the type of data being stored (e.g., integer, floating-point, character).
decoding	The integer/floating-point representation of data values passed to an application by the CDF library as they are read from a CDF. This is independent of the way the data values are physically stored in the CDF.
dimensionality	The number of dimensions and the dimension sizes for the rVariables or a zVariable.
encoding	The integer/floating-point representation of the data values physically stored in a CDF.
entry	A CDF object in which metadata is stored. An entry is associated with an attribute.
`EPOCH`	A variable of the NSSDC Standard used to store "time" values. The `CDF_EPOCH` data type is used with this variable.
error code	A status code indicating that a fatal condition was encountered. The operation was aborted.
extend records	The number of records actually allocated for a variable in a single-file CDF when a new record is written. More records may be allocated than are actually needed in order to keep a variable's records as contiguous as possible (with the assumption that the records will eventually be written by an application).
format	In reference to a CDF, the way in which files are used to store the CDF's control/data/metadata. This may be single-file or multi-file.

full-physical record	A variable record consisting of values exactly as physically stored in the CDF.
gAttribute	A global scoped attribute.
gEntry	An entry for a gAttribute.
global scope	Global scope indicates that an attribute describes some property of the entire CDF.
host decoding	The decoding of the computer currently being used.
host encoding	The encoding of the computer currently being used.
hyper access	A variable access method in which multiple records/values are read/written for a variable.
IDL Interface	A set of functions callable from within IDL (Interactive Data Language) that allow access to CDFs. The CDF distribution contains an IDL interface in addition to the CDF interface built into IDL by Research Systems, Inc. (RSI — the distributors of IDL).
IEEE 754	The floating-point representation of XDR.
include file	A file, included by a C or Fortran application, that contains constants recognized by the CDF library pertaining to various aspects of CDF objects/states.
indexing	The method used in a single-file CDF to keep track of where each variable's records are located.
informational code	A status code indicating success but providing some additional information that may be of interest.
Internal Interface	A set of routines in the CDF library callable from C and Fortran applications that provide all types of access to CDFs.
item	When the Internal Interface is used, an object or state on which an operation is performed.
`libcdf.a`	The CDF library on UNIX systems.
`libcdf.lib`	The CDF library on MS-DOS systems.
`LIBCDF.OLB`	The CDF library on VMS and OpenVMS systems.
little-endian	The byte ordering in which the least significant byte (LSB) is stored in the lowest memory location.
majority	The order in which the values of a multidimensional array are stored. This may be either row-major or column-major.
monotonicity	The property of a variable that specifies whether or not that variable's values increment or decrement (or neither) along a dimension or from record to record.
multi-file	A CDF format. Multi-file CDFs consist of one file for control/metadata and one file per variable of data.

multiple variable access	A variable access method in which one full-physical record is read/written for each of one or more variables.
network encoding	The encoding that uses the XDR representation.
`NOVARY`	A record/dimension variance indicating that the values do not change from record to record or along a dimension.
NRV variable	Non-record variant variable. A variable whose values do not change from record to record (a record variance of `NOVARY`).
NSSDC	National Space Science Data Center.
NSSDC Standard	A set of guidelines for creating a CDF containing attributes expected by NSSDC tools.
number of elements	For a variable the number of instances of the data type at each value. For an attribute entry the number of instances of the data type for that entry.
object	When the Internal Interface is used, an item that exists and may be accessed/manipulated (e.g., a CDF or variable).
Obsolete Interface	The CDF Version 1 programming interface for Fortran applications on VAX/VMS systems only. The use of this interface is discouraged.
operation	When the Internal Interface is used, an action performed on an item (e.g., creating or writing).
pad value	A value written to a variable by the CDF library in those cases where a physical record must be written but not all of its values have been specified by an application. For example, when a single value is written to a new record, all of the other values are written using the pad value.
physical record	A variable record actually stored in a CDF.
physical value	A variable value actually stored in a CDF.
read-only	A mode of the CDF library in which modifications to a CDF are not allowed.
rEntry	An entry for a vAttribute corresponding to an rVariable.
row-major	The variable majority where the last index of a multidimensional array of values increments the fastest.
RV variable	Record variant variable. A variable whose values change from record to record (a record variance of `VARY`).
rVariable	"R" variable. A CDF object in which data values are stored. All rVariables have the same dimensionality.
scope	The intended use for an attribute. This may be global scope or variable scope.

sequential access	A variable access method in which values are read/written in the physical order in which they are stored in the CDF.
single-file	A CDF format. Single-file CDFs are entirely contained within one file.
single value access	A variable access method in which exactly one value is read/written for a variable.
skeleton CDF	A CDF consisting of only control, metadata (attributes and attribute entries), and NRV variable values.
skeleton table	A text file containing the control, metadata, and NRV variable values of a skeleton CDF. A skeleton table is read by the `CDFskeleton` toolkit program, which then creates the corresponding skeleton CDF. The `SkeletonTable` toolkit program can be used to create a skeleton table from a CDF.
Standard Interface	A set of routines in the CDF library callable from C and Fortran applications that provide access to a commonly used subset of the capabilities of the Internal Interface. This interface was defined with the release of CDF V2.0 and has not changed since. New features since that time are available only through the Internal Interface (e.g., zVariables and zMode).
state	When the Internal Interface is used, a property pertaining to an object (e.g., a CDF's format or variable's data specification).
status code	The result of a CDF function/subroutine call. `CDF_OK` indicates unqualified success.
status handler	A function/subroutine that acts upon a status code received from the CDF library.
variable scope	Variable scope indicates that an attribute describes some property of each variable.
variance, record	The property of a variable that specifies whether or not its values change from record to record.
variance, dimension	The property of a variable that specifies whether or not the values along a dimension change or stay the same.
`VARY`	A record/dimension variance indicating that the values change from record to record or along a dimension.
vAttribute	A variable scoped attribute.
virtual record	A variable record that is not actually stored in a CDF but does appear in the conceptual view of the CDF. Virtual records would be those records beyond the first record of an NRV variable and those records beyond the last record actually written to an RV variable.
virtual value	A variable value this is not actually stored in a CDF but does appear in the conceptual view of the CDF. Virtual values would be those values beyond the first value of a dimension whose

	variance is `NOVARY`.
warning code	A status code indicating that the operation did complete but probably not as expected.
XDR	External Data Representation. An integer/floating-point representation using big-endian byte ordering and the IEEE 754 floating-point representation.
zEntry	An entry for a vAttribute corresponding to a zVariable.
zMode	A mode of the CDF library in which rVariables are made to appear as zVariables (and rEntries appear as zEntries).
zVariable	"Z" variable. A CDF object in which data values are stored. zVariables can have dimensionalities that are different than those of the rVariables (and each other).

Index

-

-0.0 to 0.0 mode, 29

A

ALPHAOSF1_DECODING, 111, 218
ALPHAOSF1_ENCODING, 110, 218
ALPHAVMSd_DECODING, 111, 218
ALPHAVMSd_ENCODING, 110, 217
ALPHAVMSg_DECODING, 111, 218
ALPHAVMSg_ENCODING, 110, 217
assumed scope, 54
attributes, 11, 53
    creating, 53, 122, 157, 230, 267
    current, 149, 259
        confirming, 170, 280
        selecting
            by name, 164, 274
            by number, 164, 274
    entries, 11, 55
        accessing, 55, 114, 221
        current, 149–150, 259
            confirming, 171, 281
            selecting
                by name, 165, 275
                by number, 164–165, 274–275
        data specification, 55
            changing, 128, 180, 237, 290–291
            data type, 55
                inquiring, 126, 196–197, 235, 308–309
            number of elements, 55
                inquiring, 126, 196–198, 235, 308–309
        existence, determining, 171–172, 281–282
        gEntries, 11, 55
        maximum
            inquiring, 125, 195–196, 233, 306–307
        number of
            inquiring, 195–196, 307
        numbering, 55, 108, 216
        reading, 126, 130, 196–198, 235, 238, 308–310
        rEntries, 11, 55

writing, 128, 181–182, 237, 291–292
zEntries, 11, 55
existence, determining, 170, 281
gAttributes, 11, 54
naming, 39, 53, 114, 122, 221, 230
case sensitivity, 53
inquiring, 125, 194, 233, 306
renaming, 124, 180, 232, 290
trailing blanks, 54
number of
inquiring, 118, 184, 226, 295
numbering, 54, 63, 108, 216
assigning, 54
inquiring, 123, 195, 231, 306
scopes, 54
assumed, 54, 65, 112, 220
correcting, 54, 68, 131, 179, 240, 290
occurring, 54
constants, 112, 220
GLOBAL_SCOPE, 112, 220
GLOBAL_SCOPE_ASSUMED, 112, 220
VARIABLE_SCOPE, 112, 220
VARIABLE_SCOPE_ASSUMED, 112, 220
definite, 112, 220
global, 54
inquiring, 125, 194, 233, 306
purpose, 54
restrictions, 54
variable, 54
vAttributes, 11, 54

B

big-endian, 35

C

C programming interface, 101, 114
compiling, 101
customizing, 208
linking, 103
summary, 359
typedef's, 108
CDFid, 108
CDFstatus, 108
caching scheme, files, 29
CATDESC attribute, 57

CDF
    defined, 1
CDF distribution, 77
CDF library, 2, 25
    caching scheme, 29
        selecting, 159, 161, 164, 166, 168, 170, 269, 271, 274, 276, 278, 280
    copyright notice
        length, 114, 221
        reading, 184, 294
    interfaces, 12, 25
    limits, 29
        open CDFs, 29
    modes, 27
        -0.0 to 0.0, 29
            confirming, 166, 276
            constants, 113, 220
                NEGtoPOSfp0off, 113, 221
                NEGtoPOSfp0on, 113, 221
            selecting, 113, 159, 220, 269
        decoding, 36
            confirming, 166, 276
            constants, 110, 218
                HOST_DECODING, 111, 218
                NETWORK_DECODING, 111, 218
                SUN_DECODING...VAX_DECODING, 111, 218–219
            performance considerations, 37
            selecting, 159, 269
        read-only, 27, 96
            confirming, 165, 276
            constants, 113, 220
                READONLYoff, 113, 220
                READONLYon, 113, 220
            selecting, 113, 159, 220, 269
        zMode, 27, 63, 66, 71, 77, 83, 97
            confirming, 166, 276
            constants, 113, 220
                zMODEoff, 113, 220
                zMODEon1, 113, 220
                zMODEon2, 113, 220
            example, 28
            selecting, 28, 113, 159, 220, 269
            zMode/1, 28
            zMode/2, 28
    shareable, 106, 213
    version
        inquiring, 185–186, 296–297
CDF toolkit, 12, 59
    command line syntax, 59
    default settings, 60
    executable names, 59, 62, 65, 75, 82, 85
CDF$INC, 101, 209

CDF$LIB, 103, 211
cdf.h, 101, 108
cdf.inc, 209, 216
CDF_attr_correct_scope, 240
CDF_attr_create, 230
CDF_attr_entry_inquire, 235
CDF_attr_get, 238
CDF_attr_inquire, 233
CDF_ATTR_NAME_LEN, 53, 114, 221
CDF_attr_num, 231
CDF_attr_put, 237
CDF_attr_rename, 232
CDF_close, 227
CDF_COPYRIGHT_LEN, 114, 221
CDF_create, 221
CDF_delete, 228
CDF_doc, 224
CDF_EPOCH, 38
CDF_error, 229, 317, 351
CDF_INC, 101, 209
CDF_inquire, 226
CDF_LIB, 103, 211
CDF_lib, 221, 255
CDF_MAX_DIMS, 113, 221
CDF_OK, 108, 216
CDF_open, 223
CDF_PATHNAME_LEN, 113, 221
CDF_STATUSTEXT_LEN, 114, 221
CDF_var_close, 254
CDF_var_create, 241
CDF_var_get, 249
CDF_var_hyper_get, 252
CDF_var_hyper_put, 250
CDF_var_inquire, 246
CDF_VAR_LEN, 114, 221
CDF_VAR_NAME_LEN, 41
CDF_var_num, 243
CDF_var_put, 247
CDF_var_rename, 245
CDF_WARN, 108, 216
CDFattrCorrectScope, 131
CDFattrCreate, 122
CDFattrEntryInquire, 126
CDFattrGet, 130
CDFattrInquire, 125
CDFattrNum, 123
CDFattrPut, 128
CDFattrRename, 124
CDFbrowse, 87
CDFclose, 119
CDFcompare, 61

executing, 61
output, 64
CDFconvert, 33, 36, 64
executing, 64
output, 67
CDFcreate, 114
CDFdelete, 120
CDFdir, 60
executing, 60
output, 61
CDFdoc, 117
CDFedit, 96
executing, 96
interaction with, 98
window types, 98
CDFerror, 121, 206, 351
CDFid (typedef), 108
CDFinquire, 74, 118
executing, 75
output, 78
CDFlib, 114, 146
CDFlist, 87
executing, 87
interaction with, 89
cdfmsf.inc, 210
CDFopen, 116
CDFs
accessing, 27, 30, 116, 120, 223, 228
browsing, 96, 113, 220
closing, 31, 115–116, 119, 158, 222, 224, 227, 268
comparing, 61
converting, 64
copyright notice, 0, 74, 117, 224
length, 114, 221
reading, 117, 183, 224, 294
corrupted, 114, 120, 222, 228
creating, 31, 114, 154, 221, 264
current, 149, 258
confirming, 165, 275
selecting, 158, 268
deleting, 66, 86, 114, 120, 158, 222, 228, 268
encoding, 3, 33
changing, 33, 64, 172, 282
constants, 110, 217
HOST_ENCODING, 110, 217
NETWORK_ENCODING, 110, 217
SUN_ENCODING...VAX_ENCODING, 110, 217–218
default, 155, 265
equivalent, 35
host, 34
inquiring, 118, 183, 226, 294

network, 34
performance considerations, 36
files
directory listing, 60
extensions, 32
filtering, 87
format, 2, 32
changing, 32, 64, 172, 282
constants, 109, 216
MULTI_FILE, 109, 216
SINGLE_FILE, 109, 216
default, 32, 155, 265
inquiring, 183, 294
multi-file, 32
performance considerations, 33
single-file, 32
inquiring, 74
limits, 39
listing, 87
modifying, 27, 71, 96
naming, 31, 39, 85, 113, 115–116, 221–223
trailing blanks, 31
wildcards, 59
opening, 31, 116, 157, 223, 267
overwriting, 114, 222
skeleton, 65
statistics for, 69
subsetting, 87
updating, 70
verifying, 61
version, 117, 224
inquiring, 117, 185, 224, 296
CDFscope, 68, 132, 240
executing, 68
interaction with, 68
output, 69
CDFskeleton, 32, 34, 82, 84, 321
executing, 85
CDFstats, 69
executing, 70
output, 72
CDFstatus (typedef), 108, 114
CDFvarClose, 145
CDFvarCreate, 133
CDFvarGet, 140
CDFvarHyperGet, 143
CDFvarHyperPut, 141
CDFvarInquire, 137
CDFvarNum, 135
CDFvarPut, 138
CDFvarRename, 136

CDFwalk, 92
    executing, 92
    interaction with, 94
COL_MAJOR, 111, 219
compiling
    C applications, 101
    Fortran applications, 209
compute_EPOCH, 317
computeEPOCH, 206
computers, supported, 379
conceptual view, 1

D

data specification
    attribute entry, 55
    variable, 44
data types, 37
    character, 38, 45
    constants, 109, 216
        CDF_BYTE...CDF_EPOCH, 109, 216–217
    EPOCH, 38, 206, 317
    equivalent, 39, 44
    floating point, 38
        -0.0, 38, 63, 67, 72, 77, 83, 98
    integer, 37
    non-character, 45
    number of elements, 38
decoding, CDF, 36
DECSTATION_DECODING, 111, 218
DECSTATION_ENCODING, 110, 218
definitions file, 26, 101, 103, 209, 211
DEFINITIONS.COM, 101, 103, 209, 211
definitions.csh, 101, 103, 209, 211
definitions.ksh, 101, 103, 209, 211
definitions.sh, 101, 103, 209, 211
definitions.tcsh, 101, 103, 209, 211
dimensionality, variable, 41
dimensions
    limit, 39, 113, 221
    numbering, 108, 216

E

encode_EPOCH, 318
encode_EPOCH1, 318
encode_EPOCH2, 318

encodeEPOCH, 207
encodeEPOCH1, 207
encodeEPOCH2, 207
encoding, CDF, 33
EPOCH, 38
    breaking down, 206, 317
    computing, 206, 317
    encoding, 207, 318
    parsing, 207, 318
    syntax, 39
    utility routines, 114, 206, 317
        C applications
            computeEPOCH, 206
            encodeEPOCH, 207
            encodeEPOCH1, 207
            encodeEPOCH2, 207
            EPOCHbreakdown, 206
            parseEPOCH, 207
        Fortran applications
            compute_EPOCH, 317
            encode_EPOCH, 318
            encode_EPOCH1, 318
            encode_EPOCH2, 318
            EPOCH_breakdown, 317
            parse_EPOCH, 318
EPOCH variable, 57
EPOCH_breakdown, 317
EPOCHbreakdown, 206
examples
    C applications
        closing
            CDF, 120
            rVariable, 145
        correcting
            attribute scope, 132
        creating
            attribute, 122
            CDF, 115, 146
            rVariable, 134, 198
            zVariable, 199
        deleting
            CDF, 121
        inquiring
            attribute, 125
                entry, 127
                number, 124
            CDF, 117, 119
                format, 208
            explanation text, 121
            rVariable, 138
                number, 135

Internal Interface, 146, 198
interpreting
status codes, 205
opening
CDF, 116
program, 14
reading
attribute entry, 130
rVariable values
hyper, 144, 200
single, 140
zVariable values
sequential, 201
renaming
attribute, 124, 201
rVariable, 136
status handler, 205
writing
attribute
gEntry, 129
rEntry, 129, 202
rVariable values
hyper, 142
single, 139
zVariable values
multiple variable, 203
conceptual view, 9
data set, flat, 8
Fortran applications
closing
CDF, 228
rVariable, 255
correcting
attribute scope, 240
creating
attribute, 231
CDF, 223, 256
rVariable, 242, 310
zVariable, 311
deleting
CDF, 229
inquiring
attribute, 234
entry, 236
number, 232
CDF, 225, 227
explanation text, 229
rVariable, 247
number, 244
Internal Interface, 256, 310
interpreting

status codes, 316
opening
CDF, 224
program, 21
reading
attribute entry, 239
rVariable values
hyper, 254, 312
single, 249
zVariable values
sequential, 313
renaming
attribute, 233, 313
rVariable, 245
status handler, 316
writing
attribute
gEntry, 237
rEntry, 237, 314
rVariable values
hyper, 251
single, 248
zVariable values
multiple variable, 314
physical view, 10
skeleton table, 19, 333
extend records, variable, 44

F

FIELDNAM attribute, 56
FILLVAL attribute, 57, 69, 86
FORMAT attribute, 57, 71, 77, 83, 97
format, CDF, 32
Fortran programming interface, 209, 221
argument passing, 215
compiling, 209
linking, 211
summary, 369
function prototypes, 114

G

GLOBAL_SCOPE, 112, 220
GLOBAL_SCOPE_ASSUMED, 112, 220

H

host decoding, 37
host encoding, 34
HOST_DECODING, 111, 218
HOST_ENCODING, 110, 217
HP_DECODING, 111, 218
HP_ENCODING, 110, 218
hyper access, variable, 48

I

IBMPC_DECODING, 111, 218
IBMPC_ENCODING, 110, 218
IBMRS_DECODING, 111, 218
IBMRS_ENCODING, 110, 218
IDL
    CDF's interface, 27, 339
IEEE 754, 29, 35, 38
include files, 101, 209
indexing, variable records, 32
initial records, variable, 42
interfaces, 12, 25
    IDL, 27, 339
    Internal, 13, 25, 146, 255
    mixing, 114, 221
    Obsolete, 26
    Standard, 13, 25
Internal Interface, 13, 25, 114, 146, 221, 255
    common mistakes, 204
    current objects/states, 149, 258
        attribute, 149, 259
        attribute entries, 149–150, 259
        CDF, 149, 258
        preselected, required, 154, 264
        records/dimensions, 150–152, 260–261
        sequential value, 151–152, 260, 262
        status code, 152, 262
        variables, 149, 259
    examples, 146, 198, 256, 310
    indentation/style, 153, 262
    items, 154, 264
    operations, 154, 264
        CLOSE_, 158, 268
        CONFIRM_, 165, 275
        CREATE_, 154, 264
        DELETE_, 158, 268
        GET_, 182, 293
        OPEN_, 157, 267

PUT_, 172, 282
SELECT_, 158, 268
status codes, returned, 152, 262
syntax, 153, 263
argument list, 153, 263
limitations, 153, 263
item referencing, 108, 216

L

libcdf.a, 103, 211
libcdf.lib, 103, 211
LIBCDF.OLB, 103, 211
limits, 29, 39, 113, 221
attribute names, 114, 221
copyright text, 114, 221
dimensions, 39, 113, 221
explanation text, 114, 221
IBM PC, 113, 221
Internal Interface, 153, 263
open CDFs, 29
pathnames, 113, 221
variable names, 39, 114, 221
linking
C applications, 103
Fortran applications, 211
Obsolete Interface, 26
little-endian, 35

M

majority
variable, 46
memory models (MS-DOS), 102, 105, 211, 213
MODS attribute, 57
MONOTON attribute, 57, 70
multi-file format, 32
MULTI_FILE, 109, 216
multiple variable access, 51

N

NEGtoPOSfp0off, 113, 221
NEGtoPOSfp0on, 113, 221
network encoding, 34