| | |
|---|---|
| | or zEntries. [Error] |
| `ILLEGAL_IN_zMODE` | The attempted operation is illegal while in zMode. Most operations involving rVariables or rEntries will be illegal. [Error] |
| `ILLEGAL_ON_V1_CDF` | The specified operation (i.e., opening) is not allowed on Version 1 CDFs. [Error] |
| `MULTI_FILE_FORMAT` | The specified operation is not applicable to CDFs with the multi-file format. For example, it does not make sense to inquire indexing statistics for a variable in a multi-file CDF (indexing is only used in single-file CDFs). [Informational] |
| `NA_FOR_VARIABLE` | The attempted operation is not applicable to the given variable. [Warning] |
| `NEGATIVE_FP_ZERO` | One or more of the values read/written are `-0.0` (an illegal value on VAXes and DEC Alphas running OpenVMS). [Warning] |
| `NO_ATTR_SELECTED` | An attribute has not yet been selected. First select the attribute on which to perform the operation. [Error] |
| `NO_CDF_SELECTED` | A CDF has not yet been selected. First select the CDF on which to perform the operation. [Error] |
| `NO_DELETE_ACCESS` | Deleting is not allowed (read-only access). Make sure that delete access is allowed on the CDF file(s). [Error] |
| `NO_ENTRY_SELECTED` | An attribute entry has not yet been selected. First select the entry number on which to perform the operation. [Error] |
| `NO_MORE_ACCESS` | Further access to the CDF is not allowed because of a severe error. If the CDF was being modified, an attempt was made to save the changes made prior to the severe error. In any event, the CDF should still be closed. [Error] |
| `NO_PADVALUE_SPECIFIED` | A pad value has not yet been specified. The default pad value is currently being used for the variable. The default pad value was returned. [Informational] |
| `NO_STATUS_SELECTED` | A CDF status code has not yet been selected. First select the status code on which to perform the operation. [Error] |
| `NO_SUCH_ATTR` | The named attribute was not found. Note that attribute names are case-sensitive. [Error] |
| `NO_SUCH_CDF` | The specified CDF does not exist. Check that the pathname specified is correct. [Error] |
| `NO_SUCH_ENTRY` | No such entry for specified attribute. [Error] |
| `NO_SUCH_RECORD` | The specified record does not exist for the given variable. [Error] |
| `NO_SUCH_VAR` | The named variable was not found. Note that variable names are case-sensitive. [Error] |
| `NO_VAR_SELECTED` | A variable has not yet been selected. First select the variable |

| | |
|---|---|
| | on which to perform the operation. [Error] |
| NO_VARS_IN_CDF | This CDF contains no rVariables. The operation performed is not applicable to a CDF with no rVariables. [Informational] |
| NO_WRITE_ACCESS | Write access is not allowed on the CDF file(s). Make sure that the CDF file(s) have the proper file system privileges and ownership. [Error] |
| NOT_A_CDF | Named CDF is corrupted or not actually a CDF. This can also occur if an older CDF distribution is being used to read a CDF created by a more recent CDF distribution. Contact CDF User Support if you are sure that the specified file is a CDF that should be readable by the CDF distribution being used. CDF is backward compatible but not forward compatible. [Error] |
| PRECEEDING_RECORDS_ALLOCATED | Because of the type of variable, records preceeding the range of records being allocated were automatically allocated as well. [Informational] |
| READ_ONLY_DISTRIBUTION | Your CDF distribution has been built to allow only read access to CDFs. Check with your system manager if you require write access. [Error] |
| READ_ONLY_MODE | The CDF is in read-only mode — modifications are not allowed. [Error] |
| SCRATCH_CREATE_ERROR | Cannot create a scratch file — error from file system. If a scratch directory has been specified, ensure that it is writable. [Error] |
| SCRATCH_DELETE_ERROR | Cannot delete a scratch file — error from file system. [Error] |
| SCRATCH_READ_ERROR | Cannot read from a scratch file — error from file system. [Error] |
| SCRATCH_WRITE_ERROR | Cannot write to a scratch file — error from file system. [Error] |
| SINGLE_FILE_FORMAT | The specified operation is not applicable to CDFs with the single-file format. For example, it does not make sense to close a variable in a single-file CDF. [Informational] |
| SOME_ALREADY_ALLOCATED | Some of the records being allocated were already allocated. [Informational] |
| TOO_MANY_PARMS | A type of sparse arrays or compression was encountered having too many parameters. This could be causes by a corrupted CDF or if the CDF was created/modified by a CDF distribution more recent than the one being used. [Error] |
| TOO_MANY_VARS | A multi-file CDF on a PC may contain only a limited number of variables because of the 8.3 file naming convention of MS-DOS. This consists of 100 rVariables and 100 zVariables. [Error] |
| UNKNOWN_COMPRESSION | An unknown type of compression was specified or encountered. [Error] |

| | |
|---|---|
| `UNKNOWN_SPARSENESS` | An unknown type of sparseness was specified or encountered. [Error] |
| `UNSUPPORTED_OPERATION` | The attempted operation is not supported at this time. [Error] |
| `VAR_ALREADY_CLOSED` | The specified variable is already closed. [Informational] |
| `VAR_CLOSE_ERROR` | Error detected while trying to close variable file. Check that sufficient disk space exists for the variable file and that it has not been corrupted. [Error] |
| `VAR_CREATE_ERROR` | An error occurred while creating a variable file in a multi-file CDF. Check that a file quota has not been reached. [Error] |
| `VAR_DELETE_ERROR` | An error occurred while deleting a variable file in a multi-file CDF. Check that sufficient privilege exist to delete the CDF files. [Error] |
| `VAR_EXISTS` | Named variable already exists - cannot create or rename. Each variable in a CDF must have a unique name (rVariables and zVariables can not share names). Note that trailing blanks are ignored by the CDF library when comparing variable names. [Error] |
| `VAR_NAME_TRUNC` | Variable name truncated to `CDF_VAR_NAME_LEN` characters. The variable was created but with a truncated name. [Warning] |
| `VAR_OPEN_ERROR` | An error occurred while opening variable file. Check that sufficient privilege exists to open the variable file. Also make sure that the associated variable file exists. [Error] |
| `VAR_READ_ERROR` | Failed to read variable as requested — error from file system. Check that the associated file is not corrupted. [Error] |
| `VAR_WRITE_ERROR` | Failed to write variable as requested — error from file system. Check that the associated file is not corrupted. [Error] |
| `VIRTUAL_RECORD_DATA` | One or more of the records are virtual (never actually written to the CDF). Virtual records do not physically exist in the CDF file(s) but are part of the conceptual view of the data provided by the CDF library. Virtual records are described in the Concepts chapter in the CDF User's Guide. [Informational] |

# Appendix B

# C Programming Summary

## B.1 Standard Interface

```
CDFstatus CDFcreate (CDFname, numDims, dimSizes, encoding, majority, id)
char  *CDFname;                                                    /* in  */
long  numDims;                                                     /* in  */
long  dimSizes[];                                                  /* in  */
long  encoding;                                                    /* in  */
long  majority;                                                    /* in  */
CDFid *id;                                                         /* out */

CDFstatus CDFopen (CDFname, id)
char  *CDFname;                                                    /* in  */
CDFid *id;                                                         /* out */

CDFstatus CDFdoc (id, version, release, text)
CDFid id;                                                          /* in  */
long  *version;                                                    /* out */
long  *release;                                                    /* out */
char  text[CDF_DOCUMENT_LEN+1];                                    /* out */

CDFstatus CDFinquire (id, numDims, dimSizes, encoding, majority, maxRec,
                      numVars, numAttrs)
CDFid id;                                                          /* in  */
long  *numDims;                                                    /* out */
long  dimSizes[CDF_MAX_DIMS];                                      /* out */
long  *encoding;                                                   /* out */
long  *majority;                                                   /* out */
long  *maxRec;                                                     /* out */
long  *numVars;                                                    /* out */
long  *numAttrs;                                                   /* out */

CDFstatus CDFclose (id)
```

```
CDFid id;                                                       /* in  */

CDFstatus CDFdelete (id)
CDFid id;                                                       /* in  */

CDFstatus CDFerror (status, message)
CDFstatus status;                                               /* in  */
char message[CDF_STATUSTEXT_LEN+1];                             /* out */

CDFstatus CDFattrCreate (id, attrName, attrScope, attrNum)
CDFid id;                                                       /* in  */
char  *attrName;                                                /* in  */
long  attrScope;                                                /* in  */
long  *attrNum;                                                 /* out */

long CDFattrNum (id, attrName)
CDFid id;                                                       /* in  */
char  *attrName;                                                /* in  */

CDFstatus CDFattrRename (id, attrNum, attrName)
CDFid id;                                                       /* in  */
long  attrNum;                                                  /* in  */
char  *attrName;                                                /* in  */

CDFstatus CDFattrInquire (id, attrNum, attrName, attrScope, maxEntry)
CDFid id;                                                       /* in  */
long  attrNum;                                                  /* in  */
char  *attrName;                                                /* out */
long  *attrScope;                                               /* out */
long  *maxEntry;                                                /* out */

CDFstatus CDFattrEntryInquire (id, attrNum, entryNum, dataType,
                               numElements)
CDFid id;                                                       /* in  */
long  attrNum;                                                  /* in  */
long  entryNum;                                                 /* in  */
long  *dataType;                                                /* out */
long  *numElements;                                             /* out */

CDFstatus CDFattrPut (id, attrNum, entryNum, dataType, numElements,
                      value)
CDFid id;                                                       /* in  */
long  attrNum;                                                  /* in  */
long  entryNum;                                                 /* in  */
long  dataType;                                                 /* in  */
long  numElements;                                              /* in  */
void  *value;                                                   /* in  */

CDFstatus CDFattrGet (id, attrNum, entryNum, value)
CDFid id;                                                       /* in  */
long  attrNum;                                                  /* in  */
```

```
long  entryNum;                                                        /* in  */
void  *value;                                                          /* out */

CDFstatus CDFvarCreate (id, varName, dataType, numElements,
                        recVariances, dimVariances, varNum)
CDFid id;                                                              /* in  */
char  *varName;                                                        /* in  */
long  dataType;                                                        /* in  */
long  numElements;                                                     /* in  */
long  recVariance;                                                     /* in  */
long  dimVariances[];                                                  /* in  */
long  *varNum;                                                         /* out */

long CDFvarNum (id, varName)
CDFid id;                                                              /* in  */
char  *varName;                                                        /* in  */

CDFstatus CDFvarRename (id, varNum, varName)
CDFid id;                                                              /* in  */
long  varNum;                                                          /* in  */
char  *varName;                                                        /* in  */

CDFstatus CDFvarInquire (id, varNum, varName, dataType, numElements,
                         recVariance, dimVariances)
CDFid id;                                                              /* in  */
long  varNum;                                                          /* in  */
char  *varName;                                                        /* out */
long  *dataType;                                                       /* out */
long  *numElements;                                                    /* out */
long  *recVariance;                                                    /* out */
long  dimVariances[CDF_MAX_DIMS];                                      /* out */

CDFstatus CDFvarPut (id, varNum, recNum, indices, value)
CDFid id;                                                              /* in  */
long  varNum;                                                          /* in  */
long  recNum;                                                          /* in  */
long  indices[];                                                       /* in  */
void  *value;                                                          /* in  */

CDFstatus CDFvarGet (id, varNum, recNum, indices, value)
CDFid id;                                                              /* in  */
long  varNum;                                                          /* in  */
long  recNum;                                                          /* in  */
long  indices[];                                                       /* in  */
void  *value;                                                          /* out */

CDFstatus CDFvarHyperPut (id, varNum, recStart, recCount, recInterval,
                          indices, counts, intervals, buffer)
CDFid id;                                                              /* in  */
long  varNum;                                                          /* in  */
long  recStart;                                                        /* in  */
```

```
long  recCount;                                                   /* in  */
long  recInterval;                                                /* in  */
long  indices[];                                                  /* in  */
long  counts[];                                                   /* in  */
long  intervals[];                                                /* in  */
void  *buffer;                                                    /* in  */

CDFstatus CDFvarHyperGet (id, varNum, recStart, recCount, recInterval,
                          indices, counts, intervals, buffer)
CDFid id;                                                         /* in  */
long  varNum;                                                     /* in  */
long  recStart;                                                   /* in  */
long  recCount;                                                   /* in  */
long  recInterval;                                                /* in  */
long  indices[];                                                  /* in  */
long  counts[];                                                   /* in  */
long  intervals[];                                                /* in  */
void  *buffer;                                                    /* out */

CDFstatus CDFvarClose (id, varNum)
CDFid id;                                                         /* in  */
long  varNum;                                                     /* in  */
```

## B.2 Internal Interface

```
CDFstatus CDFlib (op, ...)
long  op;                                                         /* in  */

   CLOSE_
           CDF_
           rVAR_
           zVAR_

   CONFIRM_
           ATTR_                   long *attrNum                  /* out */
           ATTR_EXISTENCE_         char *attrName                 /* in  */
           CDF_                    CDFid *id                      /* out */
           CDF_ACCESS_
           CDF_CACHESIZE_          long *numBuffers               /* out */
           CDF_DECODING_           long *decoding                 /* out */
           CDF_NAME_               char CDFname[CDF_PATHNAME_LEN+1]
                                                                  /* out */
           CDF_NEGtoPOSfp0_MODE_   long *mode                     /* out */
           CDF_READONLY_MODE_      long *mode                     /* out */
           CDF_STATUS_             CDFstatus *status              /* out */
           CDF_zMODE_              long *mode                     /* out */
           COMPRESS_CACHESIZE_     long *numBuffers               /* out */
           CURgENTRY_EXISTENCE_
           CURrENTRY_EXISTENCE_
```

```
          CURzENTRY_EXISTENCE_
          gENTRY_                 long *entryNum              /* out */
          gENTRY_EXISTENCE_       long entryNum               /* in  */
          rENTRY_                 long *entryNum              /* out */
          rENTRY_EXISTENCE_       long entryNum               /* in  */
          rVAR_                   long *varNum                /* out */
          rVAR_CACHESIZE_         long *numBuffers            /* out */
          rVAR_EXISTENCE_         char *varName               /* in  */
          rVAR_PADVALUE_
          rVAR_RESERVEPERCENT_    long *percent               /* out */
          rVAR_SEQPOS_            long *recNum                /* out */
                                  long indices[CDF_MAX_DIMS]  /* out */
          rVARs_DIMCOUNTS_        long counts[CDF_MAX_DIMS]   /* out */
          rVARs_DIMINDICES_       long indices[CDF_MAX_DIMS]  /* out */
          rVARs_DIMINTERVALS_     long intervals[CDF_MAX_DIMS] /* out */
          rVARs_RECCOUNT_         long *recCount              /* out */
          rVARs_RECINTERVAL_      long *recInterval           /* out */
          rVARs_RECNUMBER_        long *recNum                /* out */
          STAGE_CACHESIZE_        long *numBuffers            /* out */
          zENTRY_                 long *entryNum              /* out */
          zENTRY_EXISTENCE_       long entryNum               /* in  */
          zVAR_                   long *varNum                /* out */
          zVAR_CACHESIZE_         long *numBuffers            /* out */
          zVAR_DIMCOUNTS_         long counts[CDF_MAX_DIMS]   /* out */
          zVAR_DIMINDICES_        long indices[CDF_MAX_DIMS]  /* out */
          zVAR_DIMINTERVALS_      long intervals[CDF_MAX_DIMS] /* out */
          zVAR_EXISTENCE_         char *varName               /* in  */
          zVAR_PADVALUE_
          zVAR_RECCOUNT_          long *recCount              /* out */
          zVAR_RECINTERVAL_       long *recInterval           /* out */
          zVAR_RECNUMBER_         long *recNum                /* out */
          zVAR_RESERVEPERCENT_    long *percent               /* out */
          zVAR_SEQPOS_            long *recNum                /* out */
                                  long indices[CDF_MAX_DIMS]  /* out */

CREATE_
          ATTR_                   char *attrName              /* in  */
                                  long scope                  /* in  */
                                  long *attrNum               /* out */

          CDF_                    char *CDFname               /* in  */
                                  long numDims                /* in  */
                                  long dimSizes[]             /* in  */
                                  CDFid *id                   /* out */

          rVAR_                   char *varName               /* in  */
                                  long dataType               /* in  */
                                  long numElements            /* in  */
                                  long recVary                /* in  */
                                  long dimVarys[]             /* in  */
                                  long *varNum                /* out */
```

```
          zVAR_                    char *varName                 /* in  */
                                   long dataType                 /* in  */
                                   long numElements              /* in  */
                                   long numDims                  /* in  */
                                   long dimSizes[]               /* in  */
                                   long recVary                  /* in  */
                                   long dimVarys[]               /* in  */
                                   long *varNum                  /* out */

DELETE_
          ATTR_
          CDF_
          gENTRY_
          rENTRY_
          rVAR_
          rVAR_RECORDS_            long firstRecord              /* in  */
                                   long lastRecord               /* in  */
          zENTRY_
          zVAR_
          zVAR_RECORDS_            long firstRecord              /* in  */
                                   long lastRecord               /* in  */

GET_
          ATTR_MAXgENTRY_          long *maxEntry                /* out */
          ATTR_MAXrENTRY_          long *maxEntry                /* out */
          ATTR_MAXzENTRY_          long *maxEntry                /* out */
          ATTR_NAME_               char attrName[CDF_ATTR_NAME_LEN+1]
                                                                 /* out */
          ATTR_NUMBER_             char *attrName                /* in  */
                                   long *attrNum                 /* out */
          ATTR_NUMgENTRIES_        long *numEntries              /* out */
          ATTR_NUMrENTRIES_        long *numEntries              /* out */
          ATTR_NUMzENTRIES_        long *numEntries              /* out */
          ATTR_SCOPE_              long *scope                   /* out */
          CDF_COMPRESSION_         long *cType                   /* out */
                                   long cParms[CDF_MAX_PARMS]    /* out */
                                   long *cPct                    /* out */
          CDF_COPYRIGHT_           char copyRight[CDF_COPYRIGHT_LEN+1]
                                                                 /* out */
          CDF_ENCODING_            long *encoding                /* out */
          CDF_FORMAT_              long *format                  /* out */
          CDF_INCREMENT_           long *increment               /* out */
          CDF_INFO_                char *name                    /* in  */
                                   long *cType                   /* out */
                                   long cParms[CDF_MAX_PARMS]    /* out */
                                   long *cSize                   /* out */
                                   long *uSize                   /* out */
          CDF_MAJORITY_            long *majority                /* out */
          CDF_NUMATTRS_            long *numAttrs                /* out */
          CDF_NUMgATTRS_           long *numAttrs                /* out */
```

```
CDF_NUMrVARS_          long *numVars                 /* out */
CDF_NUMvATTRS_         long *numAttrs                /* out */
CDF_NUMzVARS_          long *numVars                 /* out */
CDF_RELEASE_           long *release                 /* out */
CDF_VERSION_           long *version                 /* out */
DATATYPE_SIZE_         long dataType                 /* in  */
                       long *numBytes                /* out */
gENTRY_DATA_           void *value                   /* out */
gENTRY_DATATYPE_       long *dataType                /* out */
gENTRY_NUMELEMS_       long *numElements             /* out */
LIB_COPYRIGHT_         char copyRight[CDF_COPYRIGHT_LEN+1]
                                                     /* out */
LIB_INCREMENT_         long *increment               /* out */
LIB_RELEASE_           long *release                 /* out */
LIB_subINCREMENT_      char *subincrement            /* out */
LIB_VERSION_           long *version                 /* out */
rENTRY_DATA_           void *value                   /* out */
rENTRY_DATATYPE_       long *dataType                /* out */
rENTRY_NUMELEMS_       long *numElements             /* out */
rVAR_ALLOCATEDFROM_    long startRecord              /* in  */
                       long *nextRecord              /* out */
rVAR_ALLOCATEDTO_      long startRecord              /* in  */
                       long *lastRecord              /* out */
rVAR_BLOCKINGFACTOR_   long *blockingFactor          /* out */
rVAR_COMPRESSION_      long *cType                   /* out */
                       long cParms[CDF_MAX_PARMS]    /* out */
                       long *cPct                    /* out */
rVAR_DATA_             void *value                   /* out */
rVAR_DATATYPE_         long *dataType                /* out */
rVAR_DIMVARYS_         long dimVarys[CDF_MAX_DIMS]   /* out */
rVAR_HYPERDATA_        void *buffer                  /* out */
rVAR_MAXallocREC_      long *maxRec                  /* out */
rVAR_MAXREC_           long *maxRec                  /* out */
rVAR_NAME_             char varName[CDF_VAR_NAME_LEN+1]
                                                     /* out */
rVAR_nINDEXENTRIES_    long *numEntries              /* out */
rVAR_nINDEXLEVELS_     long *numLevels               /* out */
rVAR_nINDEXRECORDS_    long *numRecords              /* out */
rVAR_NUMallocRECS_     long *numRecords              /* out */
rVAR_NUMBER_           char *varName                 /* in  */
                       long *varNum                  /* out */
rVAR_NUMELEMS_         long *numElements             /* out */
rVAR_NUMRECS_          long *numRecords              /* out */
rVAR_PADVALUE_         void *value                   /* out */
rVAR_RECVARY_          long *recVary                 /* out */
rVAR_SEQDATA_          void *value                   /* out */
rVAR_SPARSEARRAYS_     long *sArraysType             /* out */
                       long sArraysParms[CDF_MAX_PARMS]
                                                     /* out */
                       long *sArraysPct              /* out */
rVAR_SPARSERECORDS_    long *sRecordsType            /* out */
```

```
rVARs_DIMSIZES_         long dimSizes[CDF_MAX_DIMS]    /* out */
rVARs_MAXREC_           long *maxRec                   /* out */
rVARs_NUMDIMS_          long *numDims                  /* out */
rVARs_RECDATA_          long numVars                   /* in  */
                        long varNums[]                 /* in  */
                        void *buffer                   /* out */
STATUS_TEXT_            char text[CDF_STATUSTEXT_LEN+1]
                                                       /* out */
zENTRY_DATA_            void *value                    /* out */
zENTRY_DATATYPE_        long *dataType                 /* out */
zENTRY_NUMELEMS_        long *numElements              /* out */
zVAR_ALLOCATEDFROM_     long startRecord               /* in  */
                        long *nextRecord               /* out */
zVAR_ALLOCATEDTO_       long startRecord               /* in  */
                        long *lastRecord               /* out */
zVAR_BLOCKINGFACTOR_    long *blockingFactor           /* out */
zVAR_COMPRESSION_       long *cType                    /* out */
                        long cParms[CDF_MAX_PARMS]     /* out */
                        long *cPct                     /* out */
zVAR_DATA_              void *value                    /* out */
zVAR_DATATYPE_          long *dataType                 /* out */
zVAR_DIMSIZES_          long dimSizes[CDF_MAX_DIMS]    /* out */
zVAR_DIMVARYS_          long dimVarys[CDF_MAX_DIMS]    /* out */
zVAR_HYPERDATA_         void *buffer                   /* out */
zVAR_MAXallocREC_       long *maxRec                   /* out */
zVAR_MAXREC_            long *maxRec                   /* out */
zVAR_NAME_              char varName[CDF_VAR_NAME_LEN+1]
                                                       /* out */
zVAR_nINDEXENTRIES_     long *numEntries               /* out */
zVAR_nINDEXLEVELS_      long *numLevels                /* out */
zVAR_nINDEXRECORDS_     long *numRecords               /* out */
zVAR_NUMallocRECS_      long *numRecords               /* out */
zVAR_NUMBER_            char *varName                  /* in  */
                        long *varNum                   /* out */
zVAR_NUMDIMS_           long *numDims                  /* out */
zVAR_NUMELEMS_          long *numElements              /* out */
zVAR_NUMRECS_           long *numRecords               /* out */
zVAR_PADVALUE_          void *value                    /* out */
zVAR_RECVARY_           long *recVary                  /* out */
zVAR_SEQDATA_           void *value                    /* out */
zVAR_SPARSEARRAYS_      long *sArraysType              /* out */
                        long sArraysParms[CDF_MAX_PARMS]
                                                       /* out */
                        long *sArraysPct               /* out */
zVAR_SPARSERECORDS_     long *sRecordsType             /* out */
zVARs_MAXREC_           long *maxRec                   /* out */
zVARs_RECDATA_          long numVars                   /* in  */
                        long varNums[]                 /* in  */
                        void *buffer                   /* out */
```

NULL_

```
OPEN_
          CDF_                    char *CDFname                 /* in  */
                                  CDFid *id                     /* out */

PUT_
          ATTR_NAME_              char *attrName                /* in  */
          ATTR_SCOPE_             long scope                    /* in  */
          CDF_COMPRESSION_        long cType                    /* in  */
                                  long cParms[]                 /* in  */
          CDF_ENCODING_           long encoding                 /* in  */
          CDF_FORMAT_             long format                   /* in  */
          CDF_MAJORITY_           long majority                 /* in  */
          gENTRY_DATA_            long dataType                 /* in  */
                                  long numElements              /* in  */
                                  void *value                   /* in  */
          gENTRY_DATASPEC_        long dataType                 /* in  */
                                  long numElements              /* in  */
          rENTRY_DATA_            long dataType                 /* in  */
                                  long numElements              /* in  */
                                  void *value                   /* in  */
          rENTRY_DATASPEC_        long dataType                 /* in  */
                                  long numElements              /* in  */
          rVAR_ALLOCATEBLOCK_     long firstRecord              /* in  */
                                  long lastRecord               /* in  */
          rVAR_ALLOCATERECS_      long numRecords               /* in  */
          rVAR_BLOCKINGFACTOR_    long blockingFactor           /* in  */
          rVAR_COMPRESSION_       long cType                    /* in  */
                                  long cParms[]                 /* in  */
          rVAR_DATA_              void *value                   /* in  */
          rVAR_DATASPEC_          long dataType                 /* in  */
                                  long numElements              /* in  */
          rVAR_DIMVARYS_          long dimVarys[]               /* in  */
          rVAR_HYPERDATA_         void *buffer                  /* in  */
          rVAR_INITIALRECS_       long nRecords                 /* in  */
          rVAR_NAME_              char *varName                 /* in  */
          rVAR_PADVALUE_          void *value                   /* in  */
          rVAR_RECVARY_           long recVary                  /* in  */
          rVAR_SEQDATA_           void *value                   /* in  */
          rVAR_SPARSEARRAYS_      long sArraysType              /* in  */
                                  long sArraysParms[]           /* in  */
          rVAR_SPARSERECORDS_     long sRecordsType             /* in  */
          rVARs_RECDATA_          long numVars                  /* in  */
                                  long varNums[]                /* in  */
                                  void *buffer                  /* in  */
          zENTRY_DATA_            long dataType                 /* in  */
                                  long numElements              /* in  */
                                  void *value                   /* in  */
          zENTRY_DATASPEC_        long dataType                 /* in  */
                                  long numElements              /* in  */
          zVAR_ALLOCATEBLOCK_     long firstRecord              /* in  */
```

```
                          long lastRecord            /* in  */
          zVAR_ALLOCATERECS_        long numRecords            /* in  */
          zVAR_BLOCKINGFACTOR_      long blockingFactor        /* in  */
          zVAR_COMPRESSION_         long cType                 /* in  */
                                    long cParms[]              /* in  */
          zVAR_DATA_                void *value                /* in  */
          zVAR_DATASPEC_            long dataType              /* in  */
                                    long numElements           /* in  */
          zVAR_DIMVARYS_            long dimVarys[]            /* in  */
          zVAR_INITIALRECS_         long nRecords              /* in  */
          zVAR_HYPERDATA_           void *buffer               /* in  */
          zVAR_NAME_                char *varName              /* in  */
          zVAR_PADVALUE_            void *value                /* in  */
          zVAR_RECVARY_             long recVary               /* in  */
          zVAR_SEQDATA_             void *value                /* in  */
          zVAR_SPARSEARRAYS_        long sArraysType           /* in  */
                                    long sArraysParms[]        /* in  */
          zVAR_SPARSERECORDS_       long sRecordsType          /* in  */
          zVARs_RECDATA_            long numVars               /* in  */
                                    long varNums[]             /* in  */
                                    void *buffer               /* in  */

SELECT_
          ATTR_                     long attrNum               /* in  */
          ATTR_NAME_                char *attrName             /* in  */
          CDF_                      CDFid id                   /* in  */
          CDF_CACHESIZE_            long numBuffers            /* in  */
          CDF_DECODING_             long decoding              /* in  */
          CDF_NEGtoPOSfp0_MODE_     long mode                  /* in  */
          CDF_READONLY_MODE_        long mode                  /* in  */
          CDF_SCRATCHDIR_           char *dirPath              /* in  */
          CDF_STATUS_               CDFstatus status           /* in  */
          CDF_zMODE_                long mode                  /* in  */
          COMPRESS_CACHESIZE_       long numBuffers            /* in  */
          gENTRY_                   long entryNum              /* in  */
          rENTRY_                   long entryNum              /* in  */
          rENTRY_NAME_              char *varName              /* in  */
          rVAR_                     long varNum                /* in  */
          rVAR_CACHESIZE_           long numBuffers            /* in  */
          rVAR_NAME_                char *varName              /* in  */
          rVAR_RESERVEPERCENT_      long percent               /* in  */
          rVAR_SEQPOS_              long recNum                /* in  */
                                    long indices[]             /* in  */
          rVARs_CACHESIZE_          long numBuffers            /* in  */
          rVARs_DIMCOUNTS_          long counts[]              /* in  */
          rVARs_DIMINDICES_         long indices[]             /* in  */
          rVARs_DIMINTERVALS_       long intervals[]           /* in  */
          rVARs_RECCOUNT_           long recCount              /* in  */
          rVARs_RECINTERVAL_        long recInterval           /* in  */
          rVARs_RECNUMBER_          long recNum                /* in  */
          STAGE_CACHESIZE_          long numBuffers            /* in  */
```

```
zENTRY_                 long entryNum                /* in  */
zENTRY_NAME_            char *varName                /* in  */
zVAR_                   long varNum                  /* in  */
zVAR_CACHESIZE_         long numBuffers              /* in  */
zVAR_DIMCOUNTS_         long counts[]                /* in  */
zVAR_DIMINDICES_        long indices[]               /* in  */
zVAR_DIMINTERVALS_      long intervals[]             /* in  */
zVAR_NAME_              char *varName                /* in  */
zVAR_RECCOUNT_          long recCount                /* in  */
zVAR_RECINTERVAL_       long recInterval             /* in  */
zVAR_RECNUMBER_         long recNum                  /* in  */
zVAR_RESERVEPERCENT_    long percent                 /* in  */
zVAR_SEQPOS_            long recNum                  /* in  */
                        long indices[]               /* in  */
zVARs_CACHESIZE_        long numBuffers              /* in  */
zVARs_RECNUMBER_        long recNum                  /* in  */
```

## B.3 EPOCH Utility Routines

```
double computeEPOCH (year, month, day, hour, minute, second, msec)
long year;                                                   /* in  */
long month;                                                  /* in  */
long day;                                                    /* in  */
long hour;                                                   /* in  */
long minute;                                                 /* in  */
long second;                                                 /* in  */
long msec;                                                   /* in  */

void EPOCHbreakdown (epoch, year, month, day, hour, minute, second, msec)
double epoch;                                                /* in  */
long *year;                                                  /* out */
long *month;                                                 /* out */
long *day;                                                   /* out */
long *hour;                                                  /* out */
long *minute;                                                /* out */
long *second;                                                /* out */
long *msec;                                                  /* out */

void encodeEPOCH (epoch, epString)
double epoch;                                                /* in  */
char epString[EPOCH_STRING_LEN+1];                           /* out */

void encodeEPOCH1 (epoch, epString)
double epoch;                                                /* in  */
char epString[EPOCH1_STRING_LEN+1];                          /* out */

void encodeEPOCH2 (epoch, epString)
double epoch;                                                /* in  */
char epString[EPOCH2_STRING_LEN+1];                          /* out */
```

```
void encodeEPOCH3 (epoch, epString)
double epoch;                                                       /* in  */
char epString[EPOCH3_STRING_LEN+1];                                 /* out */

void encodeEPOCHx (epoch, format, epString)
double epoch;                                                       /* in  */
char format[EPOCHx_FORMAT_MAX+1];                                   /* in  */
char epString[EPOCHx_STRING_MAX+1];                                 /* out */

double parseEPOCH (epString)
char epString[EPOCH_STRING_LEN+1];                                  /* in  */

double parseEPOCH1 (epString)
char epString[EPOCH1_STRING_LEN+1];                                 /* in  */

double parseEPOCH2 (epString)
char epString[EPOCH2_STRING_LEN+1];                                 /* in  */

double parseEPOCH3 (epString)
char epString[EPOCH3_STRING_LEN+1];                                 /* in  */
```

# Index


ALPHAOSF1_DECODING, 18
ALPHAOSF1_ENCODING, 17
ALPHAVMSd_DECODING, 18
ALPHAVMSd_ENCODING, 17
ALPHAVMSg_DECODING, 18
ALPHAVMSg_ENCODING, 17
ALPHAVMSi_DECODING, 18
ALPHAVMSi_ENCODING, 17
attributes
- creating, 31, 73
- current, 58
  - confirming, 64
  - selecting
    - by name, 111
    - by number, 111
- deleting, 75
- entries
  - accessing, 23
  - current, 58–59
    - confirming, 66–67, 70
    - selecting
      - by name, 113, 116
      - by number, 113, 116
  - data specification
    - changing, 37, 100–101, 106
    - data type
      - inquiring, 35, 82, 84, 91
    - number of elements
      - inquiring, 35, 82, 84, 91
  - deleting, 75–76
  - existence, determining, 66–67, 70
  - maximum
    - inquiring, 34, 77
  - number of
    - inquiring, 78–79
  - numbering, 15
  - reading, 36, 39, 82, 84, 91
  - writing, 37, 100–101, 105
- existence, determining, 64
- naming, 22, 31
  - inquiring, 34, 78
  - renaming, 33, 99
- number of
  - inquiring, 27, 81
- numbering, 15
    - inquiring, 32, 78
- scopes
  - changing, 99
  - constants, 21
    - GLOBAL_SCOPE, 21
    - VARIABLE_SCOPE, 21
  - inquiring, 34, 79

C programming interface
- summary, 145

CDF library
- copyright notice
  - length, 22
  - reading, 83
- interfaces, 23
- modes
  - -0.0 to 0.0
    - confirming, 65
    - constants, 22
      - NEGtoPOSfp0off, 22
      - NEGtoPOSfp0on, 22
    - selecting, 22, 111
  - decoding
    - confirming, 65
    - constants, 18
      - ALPHAOSF1_DECODING, 18
      - ALPHAVMSd_DECODING, 18
      - ALPHAVMSg_DECODING, 18
      - ALPHAVMSi_DECODING, 18
      - DECSTATION_DECODING, 18
      - HOST_DECODING, 18
      - HP_DECODING, 18
      - IBMRS_DECODING, 18
      - MAC_DECODING, 18
      - NETWORK_DECODING, 18
      - NeXT_DECODING, 18
      - PC_DECODING, 18
      - SGi_DECODING, 18
      - SUN_DECODING, 18
      - VAX_DECODING, 18
    - selecting, 111
  - read-only
    - confirming, 65
    - constants, 21
      - READONLYoff, 21
      - READONLYon, 21
    - selecting, 21, 112
  - zMode
    - confirming, 65
    - constants, 21

zMODEoff, 21
zMODEon1, 22
zMODEon2, 22
selecting, 21, 112
shareable, 11
version
inquiring, 83
CDF$INC, 1
CDF$LIB, 5
cdf.h, 1, 15
CDF_ATTR_NAME_LEN, 22
CDF_BYTE, 16
CDF_CHAR, 16
CDF_COPYRIGHT_LEN, 22
CDF_DOUBLE, 16
CDF_EPOCH, 16
CDF_error, 135
CDF_FLOAT, 16
CDF_INC, 1
CDF_INT1, 16
CDF_INT2, 16
CDF_INT4, 16
CDF_LIB, 5
CDF_MAX_DIMS, 22
CDF_MAX_PARMS, 22
CDF_OK, 16
CDF_PATHNAME_LEN, 22
CDF_REAL4, 16
CDF_REAL8, 16
CDF_STATUSTEXT_LEN, 22
CDF_UCHAR, 16
CDF_UINT1, 16
CDF_UINT2, 16
CDF_UINT4, 16
CDF_VAR_LEN, 22
CDF_WARN, 16
CDFattrCreate, 31
CDFattrEntryInquire, 35
CDFattrGet, 39
CDFattrInquire, 34
CDFattrNum, 32
CDFattrPut, 37
CDFattrRename, 33
CDFclose, 28
CDFcreate, 23
CDFdelete, 29
CDFdoc, 26
CDFerror, 30, 128, 135
CDFid (typedef), 15
CDFinquire, 27
CDFlib, 55
CDFopen, 25
CDFs
accessing, 25, 29, 64
browsing, 21
cache buffers
confirming, 64, 66–67, 70
selecting, 111–114, 116, 118
closing, 24–25, 28, 63
compression
inquiring, 79–80
specifying, 99
types/parameters, 20
copyright notice, 0, 26
length, 22
reading, 26, 79
corrupted, 23, 29
creating, 23, 73
current, 58
confirming, 64–65
selecting, 111
deleting, 23, 29, 75
encoding
changing, 99
constants, 17
ALPHAOSF1_ENCODING, 17
ALPHAVMSd_ENCODING, 17
ALPHAVMSg_ENCODING, 17
ALPHAVMSi_ENCODING, 17
DECSTATION_ENCODING, 17
HOST_ENCODING, 17
HP_ENCODING, 17
IBMRS_ENCODING, 17
MAC_ENCODING, 17
NETWORK_ENCODING, 17
NeXT_ENCODING, 17
PC_ENCODING, 17
SGi_ENCODING, 17
SUN_ENCODING, 17
VAX_ENCODING, 17
default, 73
inquiring, 27, 79
format
changing, 99
constants, 16
MULTI_FILE, 16
SINGLE_FILE, 16
default, 73
inquiring, 80
naming, 22–23, 25
nulling, 98
opening, 25, 98

    overwriting, 23
    scratch directory
        specifying, 112
    version, 26
        inquiring, 26, 80–82
CDFstatus (typedef), 15, 23
CDFvarClose, 52
CDFvarCreate, 40
CDFvarGet, 47
CDFvarHyperGet, 50
CDFvarHyperPut, 49
CDFvarInquire, 44
CDFvarNum, 42
CDFvarPut, 46
CDFvarRename, 43
COLUMN_MAJOR, 19
compiling, 1
compression
    CDF
        inquiring, 79–80
        specifying, 99
    types/parameters, 20
    variables
        inquiring, 85, 92
        reserve percentage
            confirming, 68, 72
            selecting, 114, 118
        specifying, 102, 107
computeEPOCH, 129

data types
    constants, 16
        CDF_BYTE, 16
        CDF_CHAR, 16
        CDF_DOUBLE, 16
        CDF_EPOCH, 16
        CDF_FLOAT, 16
        CDF_INT1, 16
        CDF_INT2, 16
        CDF_INT4, 16
        CDF_REAL4, 16
        CDF_REAL8, 16
        CDF_UCHAR, 16
        CDF_UINT1, 16
        CDF_UINT2, 16
        CDF_UINT4, 16
    EPOCH, 129
    inquiring size, 82
DECSTATION_DECODING, 18
DECSTATION_ENCODING, 17
definitions file, 1, 5
definitions.B, 1, 5
definitions.C, 1, 5
DEFINITIONS.COM, 1, 5
definitions.K, 1, 5
dimensions
    limit, 22
    numbering, 15

encodeEPOCH, 130
encodeEPOCH1, 130
encodeEPOCH2, 130
encodeEPOCH3, 131
encodeEPOCHx, 131
EPOCH
    computing, 129
    decomposing, 129
    encoding, 130–131
    parsing, 132–133
    utility routines, 23, 129
        computeEPOCH, 129
        encodeEPOCH, 130
        encodeEPOCH1, 130
        encodeEPOCH2, 130
        encodeEPOCH3, 131
        encodeEPOCHx, 131
        EPOCHbreakdown, 129
        parseEPOCH, 132
        parseEPOCH1, 132
        parseEPOCH2, 133
        parseEPOCH3, 133
EPOCHbreakdown, 129
examples
    closing
        CDF, 29
        rVariable, 53
    creating
        attribute, 31
        CDF, 24, 55
        rVariable, 41, 119
        zVariable, 120
    deleting
        CDF, 30
    inquiring
        attribute, 35
            entry, 36
            number, 33
        CDF, 26, 28
            format, 126
        explanation text, 30
        rVariable, 45
            number, 43

Internal Interface, 55, 119
interpreting
status codes, 127
opening
CDF, 25
reading
attribute entry, 39
rVariable values
hyper, 52, 120
single, 48
zVariable values
sequential, 122
renaming
attribute, 33, 122
rVariable, 44
status handler, 127
writing
attribute
gEntry, 38
rEntry, 38, 123
rVariable values
hyper, 50
single, 46
zVariable values
multiple variable, 124

function prototypes, 23

GLOBAL_SCOPE, 21

HOST_DECODING, 18
HOST_ENCODING, 17
HP_DECODING, 18
HP_ENCODING, 17

IBMRS_DECODING, 18
IBMRS_ENCODING, 17
include files, 1
interfaces
Internal, 55
mixing, 23
Standard, 23
Internal Interface, 55
common mistakes, 125
current objects/states, 58
attribute, 58
attribute entries, 58–59
CDF, 58
preselected, required, 63
records/dimensions, 59–61
sequential value, 60–61
status code, 61
variables, 58
examples, 55, 119
indentation/style, 62
items, 63
operations, 63
status codes, returned, 61
syntax, 62
argument list, 62
limitations, 62
item referencing, 15

libcdf.a, 6
libcdf.lib, 7
libcdf.o, 9
LIBCDF.OLB, 5–6
limits, 22
attribute names, 22
copyright text, 22
dimensions, 22
explanation text, 22
file names, 22
Internal Interface, 62
parameters, 22
PC, 22
variable names, 22
linking, 5
shareable CDF library, 11

MAC_DECODING, 18
MAC_ENCODING, 17
memory models (MS-DOS), 2–3, 7–8
MULTI_FILE, 16

NEGtoPOSfp0off, 22
NEGtoPOSfp0on, 22
NETWORK_DECODING, 18
NETWORK_ENCODING, 17
NeXT_DECODING, 18
NeXT_ENCODING, 17
NO_COMPRESSION, 20
NO_SPARSEARRAYS, 21
NO_SPARSERECORDS, 20
NOVARY, 19

PAD_SPARSERECORDS, 20
parseEPOCH, 132
parseEPOCH1, 132
parseEPOCH2, 133
parseEPOCH3, 133
PC_DECODING, 18

PC_ENCODING, 17
PREV_SPARSERECORDS, 21
programming interface, 15
compiling, 1
customizing, 125
linking, 5
typedef's, 15
CDFid, 15
CDFstatus, 15

READONLYoff, 21
READONLYon, 21
ROW_MAJOR, 19

scratch directory
specifying, 112
SGi_DECODING, 18
SGi_ENCODING, 17
SINGLE_FILE, 16
sparse arrays
inquiring, 89, 97
specifying, 105, 110
types/parameters, 21
sparse records
inquiring, 89, 97
specifying, 105, 110
types/parameters, 20
Standard Interface, 23
status codes, 135
constants, 16, 135
CDF_OK, 16
CDF_WARN, 16
current, 61
confirming, 65
selecting, 112
error, 135
explanation text, 135
inquiring, 30, 91, 135
length, 22
informational, 135
interpreting, 127, 135
returned, 23
status handler, example, 127
warning, 135
SUN_DECODING, 18
SUN_ENCODING, 17

VARIABLE_SCOPE, 21
variables
accessing, 23
hyper values
dimension counts
current, 59, 61
confirming, 68, 71
selecting, 115, 117
dimension indices, starting
current, 59–60
confirming, 69, 71
selecting, 115, 117
dimension intervals
current, 60–61
confirming, 69, 71
selecting, 115, 117
reading, 50, 86, 94
record count
current, 59–60
confirming, 69, 72
selecting, 115, 117
record interval
current, 59–60
confirming, 69, 72
selecting, 115, 118
record number, starting
current, 59–60
confirming, 69, 72
selecting, 115, 118
writing, 49, 103, 108
multiple variable
reading, 90, 98
record numbers
current, 59–60
selecting, 118–119
writing, 105, 110
sequential values
current value, 60–61
confirming, 68, 72
selecting, 114, 118
reading, 89, 96
writing, 104, 109
single values
dimension indices
current, 59–60
confirming, 69, 71
selecting, 115, 117
reading, 47, 85, 93
record number
current, 59–60
confirming, 69, 72
selecting, 115, 118
writing, 46, 102, 107
closing, 52, 63–64
compression

inquiring, 85, 92
reserve percentage
confirming, 68, 72
selecting, 114, 118
specifying, 102, 107
types/parameters, 20
creating, 40, 74
current, 58
confirming, 67, 70
selecting
by name, 114, 117
by number, 113, 116
data specification
changing, 102, 107
data type
inquiring, 44, 86, 93
number of elements
inquiring, 44, 88, 96
deleting, 76
dimensionality
inquiring, 27, 90, 93, 95
existence, determining, 68, 71
indices
numbering, 15
majority
changing, 100
considering, 19
constants, 19
`COLUMN_MAJOR`, 19
`ROW_MAJOR`, 19
default, 73
inquiring, 27, 80
naming, 22, 41
inquiring, 44, 87, 94
renaming, 43, 104, 109
number of, inquiring, 27, 81
numbering, 15
inquiring, 42, 88, 95
pad values
existence, 68, 71
inquiring, 88, 96
specifying, 104, 109
records
allocated
inquiring, 84–86, 88, 92, 94–95
specifying, 101–102, 106–107
blocking factor
inquiring, 85, 92
specifying, 102, 107
deleting, 76–77
indexing
inquiring, 87, 94–95
initial
writing, 103, 108
maximum
inquiring, 27, 87, 90, 94, 97
number
inquiring, 88, 96
numbering, 15
sparse
inquiring, 89, 97
specifying, 105, 110
types, 20
sparse arrays
inquiring, 89, 97
specifying, 105, 110
types/parameters, 21
variances
constants, 19
`NOVARY`, 19
`VARY`, 19
dimensional
changing, 103, 108
inquiring, 44, 86, 93
record
changing, 104, 109
inquiring, 44, 89, 96
`VARY`, 19
`VAX_DECODING`, 18
`VAX_ENCODING`, 17

`zMODEoff`, 21
`zMODEon1`, 22
`zMODEon2`, 22