`BAD_NUM_VARS`	Illegal number of variables in a record access operation. [Error]
`BAD_READONLY_MODE`	Illegal read-only mode specified. The CDF read-only modes are defined in `cdf.h` for C applications and in `cdf.inc` for Fortran applications. [Error]
`BAD_REC_COUNT`	Illegal record count specified. A record count must be at least one (1). [Error]
`BAD_REC_INTERVAL`	Illegal record interval specified. A record interval must be at least one (1). [Error]
`BAD_REC_NUM`	Record number is out of range. Record numbers must be at least zero (0) for C applications and at least one (1) for Fortran applications. Note that a valid value must be specified regardless of the record variance. [Error]
`BAD_SCOPE`	Unknown attribute scope specified. The attribute scopes are defined in `cdf.h` for C applications and in `cdf.inc` for Fortran applications. [Error]
`BAD_VAR_NAME`	Illegal variable name specified. Variable names must contain at least one character and each character must be printable. [Error]
`BAD_VAR_NUM`	Illegal variable number specified. Variable numbers must be zero (0) or greater for C applications and one (1) or greater for Fortran applications. [Error]
`BAD_zMODE`	Illegal zMode specified. The CDF zModes are defined in `cdf.h` for C applications and in `cdf.inc` for Fortran applications. [Error]
`CANNOT_CHANGE`	Because of dependencies on the value, it cannot be changed. Some possible causes of this error follow:

1. Changing a CDF's data encoding after a variable value (including a pad value) or an attribute entry has been written.
2. Changing a CDF's format after a variable has been created.
3. Changing a CDF's variable majority after a variable value (excluding a pad value) has been written.
4. Changing a variable's data specification after a value (including the pad value) has been written to that variable.
5. Changing a variable's record variance after a value (excluding the pad value) has been written to that variable.
6. Changing a variable's dimension variances after a value (excluding the pad value) has been written

to that variable.

7. Allocating or writing "initial" records to a variable after a value (excluding the pad value) has been written to that variable.
8. Changing an attribute entry's data specification where the new specification is not equivalent to the old specification.

[Error]

`CDF_CLOSE_ERROR`	Error detected while trying to close CDF. Check that sufficient disk space exists for the `.cdf` file and that it has not been corrupted. [Error]
`CDF_CREATE_ERROR`	Cannot create CDF specified — error from file system. Make sure sure that sufficient privilege exists to create the `.cdf` file in the disk/directory location specified and that an open file quota has not already been reached. [Error]
`CDF_EXISTS`	The CDF named already exists — cannot create it. The CDF library will not overwrite an existing CDF. [Error]
`CDF_INTERNAL_ERROR`	An unexpected condition has occurred in the CDF library. Report this error to CDFsupport. [Error]
`CDF_NAME_TRUNC`	CDF pathname truncated to `CDF_PATHNAME_LEN` characters. The CDF was created but with a truncated name. [Warning]
`CDF_OK`	Function completed successfully.
`CDF_OPEN_ERROR`	Cannot open CDF specified — error from file system. Check that the `.cdf` file is not corrupted and that sufficient privilege exists to open it. Also check that an open file quota has not already been reached. [Error]
`CDF_READ_ERROR`	Failed to read CDF file — error from file system. Check that the `.cdf` file is not corrupted. [Error]
`CDF_WRITE_ERROR`	Failed to write CDF file — error from file system. Check that the `.cdf` file is not corrupted. [Error]
`CORRUPTED_V2_CDF`	This Version 2 CDF is corrupted. An error has been detected in the CDF's control information. If the CDF file(s) are known to be valid, please contact CDF User Support. [Error]
`END_OF_VAR`	The sequential access current value is at the end of the variable. Reading beyond the end of the last physical value for a variable is not allowed (when performing sequential access). [Error]
`FUTURE_CDF`	The CDF distribution being used is older than the CDF distribution which created/modified the CDF. In this case, there is no guarantee that the CDF will be successfully accessed. [Warning]

IBM_PC_OVERFLOW	An operation involving a buffer greater than 64k bytes in size has been specified. [Error]
ILLEGAL_FOR_SCOPE	The operation is illegal for the attribute's scope. For example, only gEntries may be written for gAttributes — not rEntries 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]
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, closing the CDF may save some of the modifications made since the CDF was opened. [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_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. Contact CDF User Support if you are sure that the specified file is a CDF. [Error]
`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]
`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]
`TOO_MANY_VARS`	A multi-file CDF on an IBM 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]
`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_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 (CDFpath, numDims, dimSizes, encoding, majority, id)
char  *CDFpath;                                                  /* in  */
long  numDims;                                                   /* in  */
long  dimSizes[];                                                /* in  */
long  encoding;                                                  /* in  */
long  majority;                                                  /* in  */
CDFid *id;                                                       /* out */

CDFstatus CDFopen (CDFpath, id)
char  *CDFpath;                                                  /* 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_CACHESIZE_            long *numBuffers            /* out */
              CDF_DECODING_             long *decoding              /* out */
              CDF_NAME_                 char *CDFpath               /* out */
              CDF_NEGtoPOSfp0_MODE_     long *mode                  /* out */
              CDF_READONLY_MODE_        long *mode                  /* out */
              CDF_zMODE_                long *mode                  /* 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_SEQPOS_              long *recNum              /* out */
                                    long indices[]            /* out */
          rVARs_DIMCOUNTS_          long counts[]             /* out */
          rVARs_DIMINDICES_         long indices[]            /* out */
          rVARs_DIMINTERVALS_       long intervals[]          /* out */
          rVARs_RECCOUNT_           long *recCount            /* out */
          rVARs_RECINTERVAL_        long *recInterval         /* out */
          rVARs_RECNUMBER_          long *recNum              /* out */
          zENTRY_                   long *entryNum            /* out */
          zENTRY_EXISTENCE_         long entryNum             /* in  */
          zVAR_                     long *varNum              /* out */
          zVAR_CACHESIZE_           long *numBuffers          /* out */
          zVAR_DIMCOUNTS_           long counts[]             /* out */
          zVAR_DIMINDICES_          long indices[]            /* out */
          zVAR_DIMINTERVALS_        long intervals[]          /* out */
          zVAR_EXISTENCE_           char *varName             /* in  */
          zVAR_RECCOUNT_            long *recCount            /* out */
          zVAR_RECINTERVAL_         long *recInterval         /* out */
          zVAR_RECNUMBER_           long *recNum              /* out */
          zVAR_SEQPOS_              long *recNum              /* out */
                                    long indices[]            /* out */

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

          CDF_                      char *CDFpath             /* 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_
          zENTRY_
          zVAR_

GET_
          ATTR_MAXgENTRY_        long *maxEntry               /* out */
          ATTR_MAXrENTRY_        long *maxEntry               /* out */
          ATTR_MAXzENTRY_        long *maxEntry               /* out */
          ATTR_NAME_             char *attrName               /* 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_COPYRIGHT_         char *copyRight              /* out */
          CDF_ENCODING_          long *encoding               /* out */
          CDF_FORMAT_            long *format                 /* out */
          CDF_INCREMENT_         long *increment              /* 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              /* 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_DATA_             void *value                  /* out */
          rVAR_DATATYPE_         long *dataType               /* out */
          rVAR_DIMVARYS_         long dimVarys[]              /* out */
          rVAR_EXTENDRECS_       long *NextendRecs            /* out */
          rVAR_HYPERDATA_        void *buffer                 /* out */
          rVAR_MAXallocREC_      long *maxRec                 /* out */
```

```
          rVAR_MAXREC_            long *maxRec                /* out */
          rVAR_NAME_              char *varName               /* out */
          rVAR_nINDEXENTRIES_     long *numEntries            /* out */
          rVAR_nINDEXRECORDS_     long *numRecords            /* out */
          rVAR_NUMBER_            char *varName               /* in  */
                                  long *varNum                /* out */
          rVAR_NUMELEMS_          long *numElements           /* out */
          rVAR_PADVALUE_          void *value                 /* out */
          rVAR_RECVARY_           long *recVary               /* out */
          rVAR_SEQDATA_           void *value                 /* out */
          rVARs_DIMSIZES_         long dimSizes[]             /* 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                  /* out */
          zENTRY_DATA_            void *value                 /* out */
          zENTRY_DATATYPE_        long *dataType              /* out */
          zENTRY_NUMELEMS_        long *numElements           /* out */
          zVAR_DATA_              void *value                 /* out */
          zVAR_DATATYPE_          long *dataType              /* out */
          zVAR_DIMSIZES_          long dimSizes[]             /* out */
          zVAR_DIMVARYS_          long dimVarys[]             /* out */
          zVAR_EXTENDRECS_        long *NextendRecs           /* out */
          zVAR_HYPERDATA_         void *buffer                /* out */
          zVAR_MAXallocREC_       long *maxRec                /* out */
          zVAR_MAXREC_            long *maxRec                /* out */
          zVAR_NAME_              char *varName               /* out */
          zVAR_nINDEXENTRIES_     long *numEntries            /* out */
          zVAR_nINDEXRECORDS_     long *numRecords            /* out */
          zVAR_NUMBER_            char *varName               /* in  */
                                  long *varNum                /* out */
          zVAR_NUMDIMS_           long *numDims               /* out */
          zVAR_NUMELEMS_          long *numElements           /* out */
          zVAR_PADVALUE_          void *value                 /* out */
          zVAR_RECVARY_           long *recVary               /* out */
          zVAR_SEQDATA_           void *value                 /* out */
          zVARs_MAXREC_           long *maxRec                /* out */
          zVARs_RECDATA_          long numVars                /* in  */
                                  long varNums[]              /* in  */
                                  void *buffer                /* out */

OPEN_
          CDF_                    char *CDFpath               /* in  */
                                  CDFid *id                   /* out */

PUT_
          ATTR_NAME_              char *attrName              /* in  */
          ATTR_SCOPE_             long scope                  /* 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_ALLOCATERECS_        long NallocateRecs            /* in  */
        rVAR_DATA_                void *value                   /* in  */
        rVAR_DATASPEC_            long dataType                 /* in  */
                                  long numElements              /* in  */
        rVAR_DIMVARYS_            long dimVarys[]               /* in  */
        rVAR_EXTENDRECS_          long NextendRecs              /* in  */
        rVAR_HYPERDATA_           void *buffer                  /* in  */
        rVAR_INITIALRECS_         long NinitialRecs             /* in  */
        rVAR_NAME_                char *varName                 /* in  */
        rVAR_PADVALUE_            void *value                   /* in  */
        rVAR_RECVARY_             long recVary                  /* in  */
        rVAR_SEQDATA_             void *value                   /* 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_ALLOCATERECS_        long NallocateRecs            /* in  */
        zVAR_DATA_                void *value                   /* in  */
        zVAR_DATASPEC_            long dataType                 /* in  */
                                  long numElements              /* in  */
        zVAR_DIMVARYS_            long dimVarys[]               /* in  */
        zVAR_EXTENDRECS_          long NextendRecs              /* in  */
        zVAR_INITIALRECS_         long NinitialRecs             /* 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  */
        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_STATUS_            CDFstatus status      /* in  */
CDF_zMODE_             long mode             /* 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_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  */
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_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 */

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

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 */
```

Index