	[Error]
`BAD_DIM_INDEX`	One or more dimension index is out of range. A valid value must be specified regardless of the dimension variance. Note also that the combination of dimension index, count, and interval must not specify an element beyond the end of the dimension. [Error]
`BAD_DIM_INTERVAL`	Illegal dimension interval specified. Dimension intervals must be at least one (1). [Error]
`BAD_DIM_SIZE`	Illegal dimension size specified. A dimension size must be at least one (1). [Error]
`BAD_ENCODING`	Unknown data encoding specified. The CDF encodings are defined in `cdf.h` for C applications and in `cdf.inc` for Fortran applications. [Error]
`BAD_ENTRY_NUM`	Illegal attribute entry number specified. Entry numbers must be at least zero (0) for C applications and at least one (1) for Fortran applications. [Error]
`BAD_FNC_OR_ITEM`	The specified function or item is illegal. Check that the proper number of arguments are specified for each operation being performed. Also make sure that `NULL_` is specified as the last operation. [Error]
`BAD_FORMAT`	Unknown format specified. The CDF formats are defined in `cdf.h` for C applications and in `cdf.inc` for Fortran applications. [Error]
`BAD_INITIAL_RECS`	An illegal number of records to initially write has been specified. The number of initial records must be at least one (1). [Error]
`BAD_MAJORITY`	Unknown variable majority specified. The CDF variable majorities are defined in `cdf.h` for C applications and in `cdf.inc` for Fortran applications. [Error]
`BAD_MALLOC`	Unable to allocate dynamic memory — system limit reached. Contact CDF User Support if this error occurs. [Error]
`BAD_NEGtoPOSfp0_MODE`	An illegal `-0.0` to `0.0` mode was specified. The `-0.0` to `0.0` modes are defined in `cdf.h` for C applications and in `cdf.inc` for Fortran applications. [Error]
`BAD_NUM_DIMS`	The number of dimensions specified is out of the allowed range. Zero (0) through `CDF_MAX_DIMS` dimensions are allowed. If more are needed, contact CDF User Support. [Error]
`BAD_NUM_ELEMS`	The number of elements of the data type is illegal. The number of elements must be at least one (1). For variables with a non-character data type, the number of elements must always be one (1). [Error]
`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_SCRATCH_DIR	An illegal scratch directory was specified. The scratch directory must be writeable and accessable (if a relative path was specified) from the directory in which the application has been executed. [Error]
BAD_SPARSEARRAYS_PARM	An illegal sparse arrays parameter was specified. [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_ALLOCATE_RECORDS	Records cannot be allocated for the given type of variable (e.g., a compressed variable). [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 or if a compressed single-file CDF. 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 or after records have been allocated for that variable.

5. Changing a variable's record variance after a value (excluding the pad value) has been written to that variable or after records have been allocated for that variable.
6. Changing a variable's dimension variances after a value (excluding the pad value) has been written to that variable or after records have been allocated for that variable.
7. Writing "initial" records to a variable after a value (excluding the pad value) has already been written to that variable.
8. Changing a variable's blocking factor when a compressed variable and a value (excluding the pad value) has been written or when a variable with sparse records and a value has been accessed.
9. Changing an attribute entry's data specification where the new specification is not equivalent to the old specification.

`CANNOT_COMPRESS` The CDF or variable cannot be compressed. For CDFs, this occurs if the CDF has the multi-file format. For variables, this occurs if the variable is in a multi-file CDF, values have been written to the variable, or if sparse arrays have already been specified for the variable. [Error]

`CANNOT_SPARSEARRAYS` Sparse arrays cannot be specified for the variable. This occurs if the variable is in a multi-file CDF, values have been written to the variable, records have been allocated for the variable, or if compression has already been specified for the variable. [Error]

`CANNOT_SPARSERECORDS` Sparse records cannot be specified for the variable. This occurs if the variable is in a multi-file CDF, values have been written to the variable, or records have been allocated for the variable. [Error]

`CDF_CLOSE_ERROR` Error detected while trying to close CDF. Check that sufficient disk space exists for the dotCDF file and that it has not been corrupted. [Error]

`CDF_CREATE_ERROR` Cannot create the CDF specified — error from file system. Make sure sure that sufficient privilege exists to create the dotCDF file in the disk/directory location specified and that an open file quota has not already been reached. [Error]

`CDF_DELETE_ERROR` Cannot delete the CDF specified — error from file system. Unsufficient privileges exist the delete the CDF file(s). [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 the CDF specified — error from file system. Check that the dotCDF 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 the CDF file — error from file system. Check that the dotCDF file is not corrupted. [Error]
CDF_WRITE_ERROR	Failed to write the CDF file — error from file system. Check that the dotCDF file is not corrupted. [Error]
COMPRESSION_ERROR	An error occured while compressing a CDF or block of variable records. This is an internal error in the CDF library. Contact CDF User Support. [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]
DECOMPRESSION_ERROR	An error occured while decompressing a CDF or block of variable records. The most likely cause is a corrupted dotCDF file. [Error]
DID_NOT_COMPRESS	For a compressed variable, a block of records did not compress to smaller than their uncompressed size. They have been stored uncompressed. This can result if the blocking factor is set too low or if the characteristics of the data are such that the compression algorithm choosen is unsuitable. [Informational]
EMPTY_COMPRESSED_CDF	The compressed CDF being opened is empty. This will result if a program which was creating/modifying the CDF abnormally terminated. [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]
FORCED_PARAMETER	A specified parameter was forced to an acceptable value (rather than an error being returned). [Warning]
IBM_PC_OVERFLOW	An operation involving a buffer greater than 64k bytes in size has been specified for PCs running 16-bit DOS/Windows 3.*. [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]
`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

Fortran Programming Summary

B.1 Standard Interface

```
SUBROUTINE CDF_create (CDF_name, num_dims, dim_sizes, encoding, majority,
1                      id, status)
CHARACTER CDF_name                                                  ! in
INTEGER*4 num_dims                                                  ! in
INTEGER*4 dim_sizes(*)                                              ! in
INTEGER*4 encoding                                                  ! in
INTEGER*4 majority                                                  ! in
INTEGER*4 id                                                        ! out
INTEGER*4 status                                                    ! out

SUBROUTINE CDF_open (CDF_name, id, status)
CHARACTER CDF_name*(*)                                              ! in
INTEGER*4 id                                                        ! out
INTEGER*4 status                                                    ! out

SUBROUTINE CDF_doc (id, version, release, text, status)
INTEGER*4 id                                                        ! in
INTEGER*4 version                                                   ! out
INTEGER*4 release                                                   ! out
CHARACTER text*(CDF_DOCUMENT_LEN)                                   ! out
INTEGER*4 status                                                    ! out

SUBROUTINE CDF_inquire (id, num_dims, dim_sizes, encoding, majority,
1                       max_rec, num_vars, num_attrs, status)
INTEGER*4 id                                                        ! in
INTEGER*4 num_dims                                                  ! out
INTEGER*4 dim_sizes(CDF_MAX_DIMS)                                   ! out
INTEGER*4 encoding                                                  ! out
INTEGER*4 majority                                                  ! out
INTEGER*4 max_rec                                                   ! out
```

```
INTEGER*4 num_vars                                                      ! out
INTEGER*4 num_attrs                                                     ! out
INTEGER*4 status                                                        ! out

SUBROUTINE CDF_close (id, status)
INTEGER*4 id                                                            ! in
INTEGER*4 status                                                        ! out

SUBROUTINE CDF_delete (id, status)
INTEGER*4 id                                                            ! in
INTEGER*4 status                                                        ! out

SUBROUTINE CDF_error (status, text)
INTEGER*4 status                                                        ! in
CHARACTER text*(CDF_STATUSTEXT_LEN)                                     ! out

SUBROUTINE CDF_attr_create (id, attr_name, attr_scope, attr_num, status)
INTEGER*4 id                                                            ! in
CHARACTER attr_name*(*)                                                 ! in
INTEGER*4 attr_scope                                                    ! in
INTEGER*4 attr_num                                                      ! out
INTEGER*4 status                                                        ! out

INTEGER*4 CDF_attr_num (id, attr_name)
INTEGER*4 id                                                            ! in
CHARACTER attr_name*(*)                                                 ! in

SUBROUTINE CDF_attr_rename (id, attr_num, attr_name, status)
INTEGER*4 id                                                            ! in
INTEGER*4 attr_num                                                      ! in
CHARACTER attr_name*(*)                                                 ! in
INTEGER*4 status                                                        ! out

SUBROUTINE CDF_attr_inquire (id, attr_num, attr_name, attr_scope, max_entry,
1                            status)
INTEGER*4 id                                                            ! in
INTEGER*4 attr_num                                                      ! in
CHARACTER attr_name*(CDF_ATTR_NAME_LEN)                                 ! out
INTEGER*4 attr_scope                                                    ! out
INTEGER*4 max_entry                                                     ! out
INTEGER*4 status                                                        ! out

SUBROUTINE CDF_attr_entry_inquire (id, attr_num, entry_num, data_type,
1                                  num_elements, status)
INTEGER*4 id                                                            ! in
INTEGER*4 attr_num                                                      ! in
INTEGER*4 entry_num                                                     ! in
INTEGER*4 data_type                                                     ! out
INTEGER*4 num_elements                                                  ! out
INTEGER*4 status                                                        ! out
```

```
SUBROUTINE CDF_attr_put (id, attr_num, entry_num, data_type, num_elements,
1                        value, status)
INTEGER*4 id                                                        ! in
INTEGER*4 attr_num                                                  ! in
INTEGER*4 entry_num                                                 ! in
INTEGER*4 data_type                                                 ! in
INTEGER*4 num_elements                                              ! in
<type>    value                                                     ! in
INTEGER*4 status                                                    ! out

SUBROUTINE CDF_attr_get (id, attr_num, entry_num, value, status)
INTEGER*4 id                                                        ! in
INTEGER*4 attr_num                                                  ! in
INTEGER*4 entry_num                                                 ! in
<type>    value                                                     ! out
INTEGER*4 status                                                    ! out

SUBROUTINE CDF_var_create (id, var_name, data_type, num_elements,
1                          rec_variance, dim_variances, var_num, status)
INTEGER*4 id                                                        ! in
CHARACTER var_name                                                  ! in
INTEGER*4 data_type                                                 ! in
INTEGER*4 num_elements                                              ! in
INTEGER*4 rec_variance                                              ! in
INTEGER*4 dim_variances(*)                                          ! in
INTEGER*4 var_num                                                   ! out
INTEGER*4 status                                                    ! out

INTEGER*4 FUNCTION CDF_var_num (id, var_name)
INTEGER*4 id                                                        ! in
CHARACTER var_name*(*)                                              ! in

SUBROUTINE CDF_var_rename (id, var_num, var_name, status)
INTEGER*4 id                                                        ! in
INTEGER*4 var_num                                                   ! in
CHARACTER var_name*(*)                                              ! in
INTEGER*4 status                                                    ! out

SUBROUTINE CDF_var_inquire (id, var_num, var_name, data_type, num_elements,
1                           rec_variance, dim_variances, status)
INTEGER*4 id                                                        ! in
INTEGER*4 var_num                                                   ! in
CHARACTER var_name*(CDF_VAR_NAME_LEN)                               ! out
INTEGER*4 data_type                                                 ! out
INTEGER*4 num_elements                                              ! out
INTEGER*4 rec_variance                                              ! out
INTEGER*4 dim_variances(CDF_MAX_DIMS)                               ! out
INTEGER*4 status                                                    ! out

SUBROUTINE CDF_var_put (id, var_num, rec_num, indices, value, status)
INTEGER*4 id                                                        ! in
```

```
INTEGER*4 var_num                                                     ! in
INTEGER*4 rec_num                                                     ! in
INTEGER*4 indices(*)                                                  ! in
<type>    value                                                       ! in
INTEGER*4 status                                                      ! out

SUBROUTINE CDF_var_get (id, var_num, rec_num, indices, value, status)
INTEGER*4 id                                                          ! in
INTEGER*4 var_num                                                     ! in
INTEGER*4 rec_num                                                     ! in
INTEGER*4 indices(*)                                                  ! in
<type>    value                                                       ! out
INTEGER*4 status                                                      ! out

SUBROUTINE CDF_var_hyper_put (id, var_num, rec_start, rec_count, rec_interval,
1                             indices, counts, intervals, buffer, status)
INTEGER*4 id                                                          ! in
INTEGER*4 var_num                                                     ! in
INTEGER*4 rec_start                                                   ! in
INTEGER*4 rec_count                                                   ! in
INTEGER*4 rec_interval                                                ! in
INTEGER*4 indices(*)                                                  ! in
INTEGER*4 counts(*)                                                   ! in
INTEGER*4 intervals(*)                                                ! in
<type>    buffer                                                      ! in
INTEGER*4 status                                                      ! out

SUBROUTINE CDF_var_hyper_get (id, var_num, rec_start, rec_count, rec_interval,
1                             indices, counts, intervals, buffer, status)
INTEGER*4 id                                                          ! in
INTEGER*4 var_num                                                     ! in
INTEGER*4 rec_start                                                   ! in
INTEGER*4 rec_count                                                   ! in
INTEGER*4 rec_interval                                                ! in
INTEGER*4 indices(*)                                                  ! in
INTEGER*4 counts(*)                                                   ! in
INTEGER*4 intervals(*)                                                ! in
<type>    buffer                                                      ! out
INTEGER*4 status                                                      ! out

SUBROUTINE CDF_var_close (id, var_num)
INTEGER*4 id                                                          ! in
INTEGER*4 var_num                                                     ! in
```

B.2 Internal Interface

```
INTEGER*4 FUNCTION CDF_lib (fnc, ..., status)
INTEGER*4 fnc                                                         ! in
.
```

```
.
.
INTEGER*4 status                                                      ! out

   CLOSE_
             CDF_
             rVAR_
             zVAR_

   CONFIRM_
             ATTR_                     INTEGER*4 attr_num                    ! out
             ATTR_EXISTENCE_           CHARACTER attr_name*(*)               ! in
             CDF_                      INTEGER*4 id                          ! out
             CDF_ACCESS_
             CDF_CACHESIZE_            INTEGER*4 num_buffers                 ! out
             CDF_DECODING_             INTEGER*4 decoding                    ! out
             CDF_NAME_                 CHARACTER CDF_name*(CDF_PATHNAME_LEN)
                                                                             ! out
             CDF_NEGtoPOSfp0_MODE_     INTEGER*4 mode                        ! out
             CDF_READONLY_MODE_        INTEGER*4 mode                        ! out
             CDF_STATUS_               INTEGER*4 status                      ! out
             CDF_zMODE_                INTEGER*4 mode                        ! out
             COMPRESS_CACHESIZE_       INTEGER*4 num_buffers                 ! out
             CURgENTRY_EXISTENCE_
             CURrENTRY_EXISTENCE_
             CURzENTRY_EXISTENCE_
             gENTRY_                   INTEGER*4 entry_num                   ! out
             gENTRY_EXISTENCE_         INTEGER*4 entry_num                   ! in
             rENTRY_                   INTEGER*4 entry_num                   ! out
             rENTRY_EXISTENCE_         INTEGER*4 entry_num                   ! in
             rVAR_                     INTEGER*4 var_num                     ! out
             rVAR_CACHESIZE_           INTEGER*4 num_buffers                 ! out
             rVAR_EXISTENCE_           CHARACTER var_name*(*)                ! in
             rVAR_PADVALUE_
             rVAR_RESERVEPERCENT_      INTEGER*4 percent                     ! out
             rVAR_SEQPOS_              INTEGER*4 rec_num                     ! out
                                       INTEGER*4 indices(CDF_MAX_DIMS)       ! out
             rVARs_DIMCOUNTS_          INTEGER*4 counts(CDF_MAX_DIMS)        ! out
             rVARs_DIMINDICES_         INTEGER*4 indices(CDF_MAX_DIMS)       ! out
             rVARs_DIMINTERVALS_       INTEGER*4 intervals(CDF_MAX_DIMS)     ! out
             rVARs_RECCOUNT_           INTEGER*4 rec_count                   ! out
             rVARs_RECINTERVAL_        INTEGER*4 rec_interval                ! out
             rVARs_RECNUMBER_          INTEGER*4 rec_num                     ! out
             STAGE_CACHESIZE_          INTEGER*4 num_buffers                 ! out
             zENTRY_                   INTEGER*4 entry_num                   ! out
             zENTRY_EXISTENCE_         INTEGER*4 entry_num                   ! in
             zVAR_                     INTEGER*4 var_num                     ! out
             zVAR_CACHESIZE_           INTEGER*4 num_buffers                 ! out
             zVAR_DIMCOUNTS_           INTEGER*4 counts(CDF_MAX_DIMS)        ! out
             zVAR_DIMINDICES_          INTEGER*4 indices(CDF_MAX_DIMS)       ! out
             zVAR_DIMINTERVALS_        INTEGER*4 intervals(CDF_MAX_DIMS)     ! out
```

```
        zVAR_EXISTENCE_          CHARACTER var_name*(*)            ! in
        zVAR_PADVALUE_
        zVAR_RECCOUNT_           INTEGER*4 rec_count               ! out
        zVAR_RECINTERVAL_        INTEGER*4 rec_interval            ! out
        zVAR_RECNUMBER_          INTEGER*4 rec_num                 ! out
        zVAR_RESERVEPERCENT_     INTEGER*4 percent                 ! out
        zVAR_SEQPOS_             INTEGER*4 rec_num                 ! out
                                 INTEGER*4 indices(CDF_MAX_DIMS)   ! out

CREATE_
        ATTR_                    CHARACTER attr_name*(*)           ! in
                                 INTEGER*4 scope                   ! in
                                 INTEGER*4 attr_num                ! out

        CDF_                     CHARACTER CDF_name*(*)            ! in
                                 INTEGER*4 num_dims                ! in
                                 INTEGER*4 dim_sizes(*)            ! in
                                 INTEGER*4 id                      ! out

        rVAR_                    CHARACTER var_name*(*)            ! in
                                 INTEGER*4 data_type               ! in
                                 INTEGER*4 num_elements            ! in
                                 INTEGER*4 rec_vary                ! in
                                 INTEGER*4 dim_varys               ! in
                                 INTEGER*4 var_num                 ! out

        zVAR_                    CHARACTER var_name*(*)            ! in
                                 INTEGER*4 data_type               ! in
                                 INTEGER*4 num_elements            ! in
                                 INTEGER*4 num_dims                ! in
                                 INTEGER*4 dim_sizes(*)            ! in
                                 INTEGER*4 rec_vary                ! in
                                 INTEGER*4 dim_varys               ! in
                                 INTEGER*4 var_num                 ! out

DELETE_
        ATTR_
        CDF_
        gENTRY_
        rENTRY_
        rVAR_
        rVAR_RECORDS_            INTEGER*4 first_record            ! in
                                 INTEGER*4 last_record             ! in
        zENTRY_
        zVAR_
        zVAR_RECORDS_            INTEGER*4 first_record            ! in
                                 INTEGER*4 last_record             ! in

GET_
        ATTR_MAXgENTRY_          INTEGER*4 max_entry               ! out
        ATTR_MAXrENTRY_          INTEGER*4 max_entry               ! out
```

```
ATTR_MAXzENTRY_         INTEGER*4 max_entry                ! out
ATTR_NAME_              CHARACTER attr_name*(CDF_ATTR_NAME_LEN)
                                                           ! out
ATTR_NUMBER_            CHARACTER attr_name*(*)            ! in
                        INTEGER*4 attr_num                 ! out
ATTR_NUMgENTRIES_       INTEGER*4 num_entries              ! out
ATTR_NUMrENTRIES_       INTEGER*4 num_entries              ! out
ATTR_NUMzENTRIES_       INTEGER*4 num_entries              ! out
ATTR_SCOPE_             INTEGER*4 scope                    ! out
CDF_COMPRESSION_        INTEGER*4 c_type                   ! out
                        INTEGER*4 c_parms(CDF_MAX_PARMS)   ! out
                        INTEGER*4 c_pct                    ! out
CDF_COPYRIGHT_          CHARACTER copy_right*(CDF_COPYRIGHT_LEN)
                                                           ! out
CDF_ENCODING_           INTEGER*4 encoding                 ! out
CDF_FORMAT_             INTEGER*4 format                   ! out
CDF_INCREMENT_          INTEGER*4 increment                ! out
CDF_INFO_               CHARACTER CDF_name*(*)             ! in
                        INTEGER*4 c_type                   ! out
                        INTEGER*4 c_parms(CDF_MAX_PARMS)   ! out
                        INTEGER*4 c_size                   ! out
                        INTEGER*4 u_size                   ! out
CDF_MAJORITY_           INTEGER*4 majority                 ! out
CDF_NUMATTRS_           INTEGER*4 num_attrs                ! out
CDF_NUMgATTRS_          INTEGER*4 num_attrs                ! out
CDF_NUMrVARS_           INTEGER*4 num_vars                 ! out
CDF_NUMvATTRS_          INTEGER*4 num_attrs                ! out
CDF_NUMzVARS_           INTEGER*4 num_vars                 ! out
CDF_RELEASE_            INTEGER*4 release                  ! out
CDF_VERSION_            INTEGER*4 version                  ! out
DATATYPE_SIZE_          INTEGER*4 data_type                ! in
                        INTEGER*4 num_bytes                ! out
gENTRY_DATA_            <type> value                       ! out
gENTRY_DATATYPE_        INTEGER*4 data_type                ! out
gENTRY_NUMELEMS_        INTEGER*4 num_elements             ! out
LIB_COPYRIGHT_          CHARACTER copy_right*(CDF_COPYRIGHT_LEN)
                                                           ! out
LIB_INCREMENT_          INTEGER*4 increment                ! out
LIB_RELEASE_            INTEGER*4 release                  ! out
LIB_subINCREMENT_       CHARACTER subincrement*1           ! out
LIB_VERSION_            INTEGER*4 version                  ! out
rENTRY_DATA_            <type> value                       ! out
rENTRY_DATATYPE_        INTEGER*4 data_type                ! out
rENTRY_NUMELEMS_        INTEGER*4 num_elements             ! out
rVAR_ALLOCATEDFROM_     INTEGER*4 start_record             ! in
                        INTEGER*4 next_record              ! out
rVAR_ALLOCATEDTO_       INTEGER*4 start_record             ! in
                        INTEGER*4 last_record              ! out
rVAR_BLOCKINGFACTOR_    INTEGER*4 blocking_factor          ! out
rVAR_COMPRESSION_       INTEGER*4 c_type                   ! out
                        INTEGER*4 c_parms(CDF_MAX_PARMS)   ! out
```

```
                         INTEGER*4 c_pct                      ! out
rVAR_DATA_               <type> value                         ! out
rVAR_DATATYPE_           INTEGER*4 data_type                  ! out
rVAR_DIMVARYS_           INTEGER*4 dim_varys(CDF_MAX_DIMS)    ! out
rVAR_HYPERDATA_          <type> buffer                        ! out
rVAR_MAXallocREC_        INTEGER*4 max_rec                    ! out
rVAR_MAXREC_             INTEGER*4 max_rec                    ! out
rVAR_NAME_               CHARACTER var_name*(CDF_VAR_NAME_LEN)
                                                              ! out
rVAR_nINDEXENTRIES_      INTEGER*4 num_entries                ! out
rVAR_nINDEXLEVELS_       INTEGER*4 num_levels                 ! out
rVAR_nINDEXRECORDS_      INTEGER*4 num_records                ! out
rVAR_NUMallocRECS_       INTEGER*4 num_records                ! out
rVAR_NUMBER_             CHARACTER var_name*(*)               ! in
                         INTEGER*4 var_num                    ! out
rVAR_NUMELEMS_           INTEGER*4 num_elements               ! out
rVAR_NUMRECS_            INTEGER*4 num_records                ! out
rVAR_PADVALUE_           <type> value                         ! out
rVAR_RECVARY_            INTEGER*4 rec_vary                   ! out
rVAR_SEQDATA_            <type> value                         ! out
rVAR_SPARSEARRAYS_       INTEGER*4 s_arrays_type              ! out
                         INTEGER*4 s_arrays_parms(CDF_MAX_PARMS)
                                                              ! out
                         INTEGER*4 s_arrays_pct               ! out
rVAR_SPARSERECORDS_      INTEGER*4 s_records_type             ! out
rVARs_DIMSIZES_          INTEGER*4 dim_sizes(CDF_MAX_PARMS)   ! out
rVARs_MAXREC_            INTEGER*4 max_rec                    ! out
rVARs_NUMDIMS_           INTEGER*4 num_dims                   ! out
rVARs_RECDATA_           INTEGER*4 num_vars                   ! in
                         INTEGER*4 var_nums(*)                ! in
                         <type> buffer                        ! out
STATUS_TEXT_             CHARACTER text*(CDF_STATUSTEXT_LEN)  ! out
zENTRY_DATATYPE_         INTEGER*4 data_type                  ! out
zENTRY_DATA_             <type> value                         ! out
zENTRY_NUMELEMS_         INTEGER*4 num_elements               ! out
zVAR_ALLOCATEDFROM_      INTEGER*4 start_record               ! in
                         INTEGER*4 next_record                ! out
zVAR_ALLOCATEDTO_        INTEGER*4 start_record               ! in
                         INTEGER*4 last_record                ! out
zVAR_BLOCKINGFACTOR_     INTEGER*4 blocking_factor            ! out
zVAR_COMPRESSION_        INTEGER*4 c_type                     ! out
                         INTEGER*4 c_parms(CDF_MAX_PARMS)     ! out
                         INTEGER*4 c_pct                      ! out
zVAR_DATA_               <type> value                         ! out
zVAR_DATATYPE_           INTEGER*4 data_type                  ! out
zVAR_DIMSIZES_           INTEGER*4 dim_sizes(CDF_MAX_DIMS)    ! out
zVAR_DIMVARYS_           INTEGER*4 dim_varys(CDF_MAX_DIMS)    ! out
zVAR_HYPERDATA_          <type> buffer                        ! out
zVAR_MAXallocREC_        INTEGER*4 max_rec                    ! out
zVAR_MAXREC_             INTEGER*4 max_rec                    ! out
zVAR_NAME_               CHARACTER var_name*(CDF_VAR_NAME_LEN)
```

```
                                                            ! out
          zVAR_nINDEXENTRIES_     INTEGER*4 num_entries            ! out
          zVAR_nINDEXLEVELS_      INTEGER*4 num_levels             ! out
          zVAR_nINDEXRECORDS_     INTEGER*4 num_records            ! out
          zVAR_NUMallocRECS_      INTEGER*4 num_records            ! out
          zVAR_NUMBER_            CHARACTER var_name*(*)           ! in
                                  INTEGER*4 var_num                ! out
          zVAR_NUMDIMS_           INTEGER*4 num_dims               ! out
          zVAR_NUMELEMS_          INTEGER*4 num_elements           ! out
          zVAR_NUMRECS_           INTEGER*4 num_records            ! out
          zVAR_PADVALUE_          <type> value                     ! out
          zVAR_RECVARY_           INTEGER*4 rec_vary               ! out
          zVAR_SEQDATA_           <type> value                     ! out
          zVAR_SPARSEARRAYS_      INTEGER*4 s_arrays_type          ! out
                                  INTEGER*4 s_arrays_parms(CDF_MAX_PARMS)
                                                                   ! out
                                  INTEGER*4 s_arrays_pct           ! out
          zVAR_SPARSERECORDS_     INTEGER*4 s_records_type         ! out
          zVARs_MAXREC_           INTEGER*4 max_rec                ! out
          zVARs_RECDATA_          INTEGER*4 num_vars               ! in
                                  INTEGER*4 var_nums(*)            ! in
                                  <type> buffer                    ! out

NULL_
                                  INTEGER*4 status                 ! out

OPEN_
          CDF_                    CHARACTER CDF_name*(*)           ! in
                                  INTEGER*4 id                     ! out

PUT_
          ATTR_NAME_              CHARACTER attr_name*(*)          ! in
          ATTR_SCOPE_             INTEGER*4 scope                  ! in
          CDF_COMPRESSION_        INTEGER*4 c_type                 ! in
                                  INTEGER*4 c_parms(*)             ! in
          CDF_ENCODING_           INTEGER*4 encoding               ! in
          CDF_FORMAT_             INTEGER*4 format                 ! in
          CDF_MAJORITY_           INTEGER*4 majority               ! in
          gENTRY_DATA_            INTEGER*4 data_type              ! in
                                  INTEGER*4 num_elements           ! in
                                  <type> value                     ! in
          gENTRY_DATASPEC_        INTEGER*4 data_type              ! in
                                  INTEGER*4 num_elements           ! in
          rENTRY_DATA_            INTEGER*4 data_type              ! in
                                  INTEGER*4 num_elements           ! in
                                  <type> value                     ! in
          rENTRY_DATASPEC_        INTEGER*4 data_type              ! in
                                  INTEGER*4 num_elements           ! in
          rVAR_ALLOCATEBLOCK_     INTEGER*4 first_record           ! in
                                  INTEGER*4 last_record            ! in
          rVAR_ALLOCATERECS_      INTEGER*4 num_records            ! in
```

```
        rVAR_BLOCKINGFACTOR_     INTEGER*4 blocking_factor          ! in
        rVAR_COMPRESSION_        INTEGER*4 c_type                   ! in
                                 INTEGER*4 c_parms(*)               ! in
        rVAR_DATA_               <type> value                       ! in
        rVAR_DATASPEC_           INTEGER*4 data_type                ! in
                                 INTEGER*4 num_elements             ! in
        rVAR_DIMVARYS_           INTEGER*4 dim_varys(*)             ! in
        rVAR_HYPERDATA_          <type> buffer                      ! in
        rVAR_INITIALRECS_        INTEGER*4 num_records              ! in
        rVAR_NAME_               CHARACTER var_name*(*)             ! in
        rVAR_PADVALUE_           <type> value                       ! in
        rVAR_RECVARY_            INTEGER*4 rec_vary                 ! in
        rVAR_SEQDATA_            <type> value                       ! in
        rVAR_SPARSEARRAYS_       INTEGER*4 s_arrays_type            ! in
                                 INTEGER*4 s_arrays_parms(*)        ! in
        rVAR_SPARSERECORDS_      INTEGER*4 s_records_type           ! in
        rVARs_RECDATA_           INTEGER*4 num_vars                 ! in
                                 INTEGER*4 var_nums(*)              ! in
                                 <type> buffer                      ! in
        zENTRY_DATA_             INTEGER*4 data_type                ! in
                                 INTEGER*4 num_elements             ! in
                                 <type> value                       ! in
        zENTRY_DATASPEC_         INTEGER*4 data_type                ! in
                                 INTEGER*4 num_elements             ! in
        zVAR_ALLOCATEBLOCK_      INTEGER*4 first_record             ! in
                                 INTEGER*4 last_record              ! in
        zVAR_ALLOCATERECS_       INTEGER*4 num_records              ! in
        zVAR_BLOCKINGFACTOR_     INTEGER*4 blocking_factor          ! in
        zVAR_COMPRESSION_        INTEGER*4 c_type                   ! in
                                 INTEGER*4 c_parms(*)               ! in
        zVAR_DATA_               <type> value                       ! in
        zVAR_DATASPEC_           INTEGER*4 data_type                ! in
                                 INTEGER*4 num_elements             ! in
        zVAR_DIMVARYS_           INTEGER*4 dim_varys(*)             ! in
        zVAR_INITIALRECS_        INTEGER*4 num_records              ! in
        zVAR_HYPERDATA_          <type> buffer                      ! in
        zVAR_NAME_               CHARACTER var_name*(*)             ! in
        zVAR_PADVALUE_           <type> value                       ! in
        zVAR_RECVARY_            INTEGER*4 rec_vary                 ! in
        zVAR_SEQDATA_            <type> value                       ! in
        zVAR_SPARSEARRAYS_       INTEGER*4 s_arrays_type            ! in
                                 INTEGER*4 s_arrays_parms(*)        ! in
        zVAR_SPARSERECORDS_      INTEGER*4 s_records_type           ! in
        zVARs_RECDATA_           INTEGER*4 num_vars                 ! in
                                 INTEGER*4 var_nums(*)              ! in
                                 <type> buffer                      ! in

SELECT_
        ATTR_                    INTEGER*4 attr_num                 ! in
        ATTR_NAME_               CHARACTER attr_name*(*)            ! in
        CDF_                     INTEGER*4 id                       ! in
```

```
CDF_CACHESIZE_          INTEGER*4 num_buffers              ! in
CDF_DECODING_           INTEGER*4 decoding                 ! in
CDF_NEGtoPOSfp0_MODE_   INTEGER*4 mode                     ! in
CDF_READONLY_MODE_      INTEGER*4 mode                     ! in
CDF_SCRATCHDIR_         CHARACTER dir_name*(*)             ! in
CDF_STATUS_             INTEGER*4 status                   ! in
CDF_zMODE_              INTEGER*4 mode                     ! in
COMPRESS_CACHESIZE_     INTEGER*4 num_buffers              ! in
gENTRY_                 INTEGER*4 entry_num                ! in
rENTRY_                 INTEGER*4 entry_num                ! in
rENTRY_NAME_            CHARACTER var_name*(*)             ! in
rVAR_                   INTEGER*4 var_num                  ! in
rVAR_CACHESIZE_         INTEGER*4 num_buffers              ! in
rVAR_NAME_              CHARACTER var_name*(*)             ! in
rVAR_RESERVEPERCENT_    INTEGER*4 percent                  ! in
rVAR_SEQPOS_            INTEGER*4 rec_num                  ! in
                        INTEGER*4 indices(*)               ! in
rVARs_CACHESIZE_        INTEGER*4 num_buffers              ! in
rVARs_DIMCOUNTS_        INTEGER*4 counts(*)                ! in
rVARs_DIMINDICES_       INTEGER*4 indices(*)               ! in
rVARs_DIMINTERVALS_     INTEGER*4 intervals(*)             ! in
rVARs_RECCOUNT_         INTEGER*4 rec_count                ! in
rVARs_RECINTERVAL_      INTEGER*4 rec_interval             ! in
rVARs_RECNUMBER_        INTEGER*4 rec_num                  ! in
STAGE_CACHESIZE_        INTEGER*4 num_buffers              ! in
zENTRY_                 INTEGER*4 entry_num                ! in
zENTRY_NAME_            CHARACTER var_name*(*)             ! in
zVAR_                   INTEGER*4 var_num                  ! in
zVAR_CACHESIZE_         INTEGER*4 num_buffers              ! in
zVAR_NAME_              CHARACTER var_name*(*)             ! in
zVAR_DIMCOUNTS_         INTEGER*4 counts(*)                ! in
zVAR_DIMINDICES_        INTEGER*4 indices(*)               ! in
zVAR_DIMINTERVALS_      INTEGER*4 intervals(*)             ! in
zVAR_RECCOUNT_          INTEGER*4 rec_count                ! in
zVAR_RECINTERVAL_       INTEGER*4 rec_interval             ! in
zVAR_RECNUMBER_         INTEGER*4 rec_num                  ! in
zVAR_RESERVEPERCENT_    INTEGER*4 percent                  ! in
zVAR_SEQPOS_            INTEGER*4 rec_num                  ! in
                        INTEGER*4 indices(*)               ! in
zVARs_CACHESIZE_        INTEGER*4 num_buffers              ! in
zVARs_RECNUMBER_        INTEGER*4 rec_num                  ! in
```

B.3 EPOCH Utility Routines

```
SUBROUTINE compute_EPOCH (year, month, day, hour, minute, second, msec, epoch)
INTEGER*4 year                                                        ! in
INTEGER*4 month                                                       ! in
INTEGER*4 day                                                         ! in
INTEGER*4 hour                                                        ! in
```

```
INTEGER*4 minute                                                         ! in
INTEGER*4 second                                                         ! in
INTEGER*4 msec                                                           ! in
REAL*8    epoch                                                          ! out

SUBROUTINE EPOCH_breakdown (epoch, year, month, day, hour, minute, second,
1                           msec)
REAL*8    epoch                                                          ! in
INTEGER*4 year                                                           ! out
INTEGER*4 month                                                          ! out
INTEGER*4 day                                                            ! out
INTEGER*4 hour                                                           ! out
INTEGER*4 minute                                                         ! out
INTEGER*4 second                                                         ! out
INTEGER*4 msec                                                           ! out

SUBROUTINE encode_EPOCH (epoch, ep_string)
REAL*8    epoch                                                          ! in
CHARACTER ep_string*(EPOCH_STRING_LEN)                                   ! out

SUBROUTINE encode_EPOCH1 (epoch, ep_string)
REAL*8    epoch                                                          ! in
CHARACTER ep_string*(EPOCH1_STRING_LEN)                                  ! out

SUBROUTINE encode_EPOCH2 (epoch, ep_string)
REAL*8    epoch                                                          ! in
CHARACTER ep_string*(EPOCH2_STRING_LEN)                                  ! out

SUBROUTINE encode_EPOCHx (epoch, format, encoded)
REAL*8    epoch                                                          ! in
CHARACTER format(EPOCHx_FORMAT_MAX)                                      ! in
CHARACTER encoded(EPOCHx_STRING_MAX)                                     ! out

SUBROUTINE parse_EPOCH (ep_string, epoch)
CHARACTER ep_string*(EPOCH_STRING_LEN)                                   ! in
REAL*8    epoch                                                          ! out

SUBROUTINE parse_EPOCH1 (ep_string, epoch)
CHARACTER ep_string*(EPOCH1_STRING_LEN)                                  ! in
REAL*8    epoch                                                          ! out

SUBROUTINE parse_EPOCH2 (ep_string, epoch)
CHARACTER ep_string*(EPOCH2_STRING_LEN)                                  ! in
REAL*8    epoch                                                          ! out

SUBROUTINE parse_EPOCH3 (ep_string, epoch)
CHARACTER ep_string*(EPOCH3_STRING_LEN)                                  ! in
REAL*8    epoch                                                          ! out
```

Index