`ATTR_EXISTS`	Named attribute already exists — cannot create or rename. Each attribute in a CDF must have a unique name. Note that trailing blanks are ignored by the CDF library when comparing attribute names. [Error]
`ATTR_NAME_TRUNC`	Attribute name truncated to `CDF_ATTR_NAME_LEN` characters. The attribute was created but with a truncated name. [Warning]
`BAD_ALLOCATE_RECS`	An illegal number of records to allocate for a variable was specified. For RV variables the number must be one or greater. For NRV variables the number must be exactly one. [Error]
`BAD_ARGUMENT`	An illegal/undefined argument was passed. Check that all arguments are properly declared and initialized. [Error]
`BAD_ATTR_NAME`	Illegal attribute name specified. Attribute names must contain at least one character, and each character must be printable. [Error]
`BAD_ATTR_NUM`	Illegal attribute number specified. Attribute numbers must be zero (0) or greater for C applications and one (1) or greater for Fortran applications. [Error]
`BAD_CACHESIZE`	An illegal number of cache buffers was specified. The value must be at least one (1). [Error]
`BAD_CDF_EXTENSION`	An illegal file extension was specified for a CDF. In general, do not specify an extension except possibly for a single-file CDF which has been renamed with a different file extension or no file extension. [Error]
`BAD_CDF_ID`	CDF identifier is unknown or invalid. The CDF identifier specified is not for a currently open CDF. [Error]
`BAD_CDF_NAME`	Illegal CDF name specified. CDF names must contain at least one character, and each character must be printable. Trailing blanks are allowed but will be ignored. [Error]
`BAD_CDFSTATUS`	Unknown CDF status code received. The status code specified is not used by the CDF library. [Error]
`BAD_DATA_TYPE`	Unknown data type specified. The CDF data types are defined in `cdf.h` for C applications and in `cdf.inc` for Fortran applications. [Error]
`BAD_DECODING`	Unknown decoding specified. The CDF decodings are defined in `cdf.h` for C applications and in `cdf.inc` for Fortran applications. [Error]
`BAD_DIM_COUNT`	Illegal dimension count specified. A dimension count must be at least one (1) and not greater than the size of the dimension. [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 in-

terval 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_EXTEND_RECS`	An illegal number of extend records has been specified. The number of extend records must be at least zero (0). A value of zero indicates that the default should be used. Note that extend records are only significant for variables that are not NRV and are in single-file CDFs. [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 non-character string variables, 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_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

Fortran Programming Summary

B.1 Standard Interface

```
SUBROUTINE CDF_create (CDF_path, num_dims, dim_sizes, encoding, majority,
1                      id, status)
CHARACTER CDF_path                                                  ! 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_path, id, status)
CHARACTER CDF_path*(*)                                              ! 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 (op, ..., status)
INTEGER*4 op                                                            ! in
.
```

```
.
.
INTEGER*4 status                                          ! out

   CLOSE_
             CDF_
             rVAR_
             zVAR_

   CONFIRM_
             ATTR_                    INTEGER*4 attr_num                ! out
             ATTR_EXISTENCE_          CHARACTER attr_name*(*)           ! in
             CDF_CACHESIZE_           INTEGER*4 num_buffers             ! out
             CDF_DECODING_            INTEGER*4 decoding                ! out
             CDF_NAME_                CHARACTER CDF_path*(*)            ! out
             CDF_NEGtoPOSfp0_MODE_    INTEGER*4 mode                    ! out
             CDF_READONLY_MODE_       INTEGER*4 mode                    ! out
             CDF_zMODE_               INTEGER*4 mode                    ! 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_SEQPOS_             INTEGER*4 rec_num                 ! out
                                      INTEGER*4 indices(*)              ! out
             rVARs_DIMCOUNTS_         INTEGER*4 counts(*)               ! out
             rVARs_DIMINDICES_        INTEGER*4 indices(*)              ! out
             rVARs_DIMINTERVALS_      INTEGER*4 intervals(*)            ! out
             rVARs_RECCOUNT_          INTEGER*4 rec_count               ! out
             rVARs_RECINTERVAL_       INTEGER*4 rec_interval            ! out
             rVARs_RECNUMBER_         INTEGER*4 rec_num                 ! 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_EXISTENCE_          CHARACTER var_name*(*)            ! in
             zVAR_DIMCOUNTS_          INTEGER*4 counts(*)               ! out
             zVAR_DIMINDICES_         INTEGER*4 indices(*)              ! out
             zVAR_DIMINTERVALS_       INTEGER*4 intervals(*)            ! out
             zVAR_RECCOUNT_           INTEGER*4 rec_count               ! out
             zVAR_RECINTERVAL_        INTEGER*4 rec_interval            ! out
             zVAR_RECNUMBER_          INTEGER*4 rec_num                 ! out
             zVAR_SEQPOS_             INTEGER*4 rec_num                 ! out
                                      INTEGER*4 indices(*)              ! out

   CREATE_
```

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

        CDF_                   CHARACTER CDF_path*(*)            ! 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_
        zENTRY_
        zVAR_

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*(*)           ! 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_COPYRIGHT_         CHARACTER copy_right*(*)          ! out
        CDF_ENCODING_          INTEGER*4 encoding                ! out
        CDF_FORMAT_            INTEGER*4 format                  ! out
        CDF_INCREMENT_         INTEGER*4 increment               ! 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*(*)             ! 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_DATA_              <type> value                         ! out
rVAR_DATATYPE_          INTEGER*4 data_type                  ! out
rVAR_DIMVARYS_          INTEGER*4 dim_varys(*)               ! out
rVAR_EXTENDRECS_        INTEGER*4 n_extend_recs              ! 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*(*)               ! out
rVAR_nINDEXENTRIES_     INTEGER*4 num_entries                ! out
rVAR_nINDEXRECORDS_     INTEGER*4 num_records                ! out
rVAR_NUMBER_            CHARACTER var_name*(*)               ! in
                        INTEGER*4 var_num                    ! out
rVAR_NUMELEMS_          INTEGER*4 num_elements               ! out
rVAR_PADVALUE_          <type> value                         ! out
rVAR_RECVARY_           INTEGER*4 rec_vary                   ! out
rVAR_SEQDATA_           <type> value                         ! out
rVARs_DIMSIZES_         INTEGER*4 dim_sizes(*)               ! 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*(*)                   ! out
zENTRY_DATATYPE_        INTEGER*4 data_type                  ! out
zENTRY_DATA_            <type> value                         ! out
zENTRY_NUMELEMS_        INTEGER*4 num_elements               ! out
zVAR_DATA_              <type> value                         ! out
zVAR_DATATYPE_          INTEGER*4 data_type                  ! out
zVAR_DIMSIZES_          INTEGER*4 dim_sizes(*)               ! out
zVAR_DIMVARYS_          INTEGER*4 dim_varys(*)               ! out
zVAR_EXTENDRECS_        INTEGER*4 n_extend_recs              ! 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*(*)               ! out
          zVAR_nINDEXENTRIES_      INTEGER*4 num_entries                ! out
          zVAR_nINDEXRECORDS_      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_PADVALUE_           <type> value                         ! out
          zVAR_RECVARY_            INTEGER*4 rec_vary                   ! out
          zVAR_SEQDATA_            <type> value                         ! out
          zVARs_MAXREC_            INTEGER*4 max_rec                    ! out
          zVARs_RECDATA_           INTEGER*4 num_vars                   ! in
                                   INTEGER*4 var_nums(*)                ! in
                                   <type> buffer                        ! out

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

PUT_
          ATTR_NAME_               CHARACTER attr_name*(*)              ! in
          ATTR_SCOPE_              INTEGER*4 scope                      ! 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_ALLOCATERECS_       INTEGER*4 n_allocate_recs            ! 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_EXTENDRECS_         INTEGER*4 n_extend_recs              ! in
          rVAR_HYPERDATA_          <type> buffer                        ! in
          rVAR_INITIALRECS_        INTEGER*4 n_initial_recs             ! in
          rVAR_NAME_               CHARACTER var_name*(*)               ! in
          rVAR_PADVALUE_           <type> value                         ! in
          rVAR_RECVARY_            INTEGER*4 rec_vary                   ! in
          rVAR_SEQDATA_            <type> value                         ! 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_ALLOCATERECS_     INTEGER*4 n_allocate_recs              ! 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_EXTENDRECS_       INTEGER*4 n_extend_recs                ! in
          zVAR_INITIALRECS_      INTEGER*4 n_initial_recs               ! 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
          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_STATUS_            INTEGER*4 status                       ! in
          CDF_zMODE_             INTEGER*4 mode                         ! 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_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
          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_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 parse_EPOCH (ep_string, epoch)
CHARACTER ep_string*(EPOCH_STRING_LEN)                              ! in
REAL*8    epoch                                                     ! 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
```

Index

ALPHAOSF1_DECODING, 16
ALPHAOSF1_ENCODING, 16
ALPHAVMSd_DECODING, 16
ALPHAVMSd_ENCODING, 15
ALPHAVMSg_DECODING, 16
ALPHAVMSg_ENCODING, 15
attributes
    creating, 30, 72
    current, 58
        confirming, 64
        selecting
            by name, 104
            by number, 104
    deleting, 74
    entries
        accessing, 21
        current, 58–59
            confirming, 66–67, 69
            selecting
                by name, 106, 109
                by number, 106, 109
        data specification
            changing, 37, 95–96, 100
            data type
                inquiring, 35, 80, 82, 88
            number of elements
                inquiring, 35, 80, 82, 88
        deleting, 75
        existence, determining, 66–67, 69
        maximum
            inquiring, 33, 76
        number of
            inquiring, 77
        numbering, 14
        reading, 35, 38, 80, 82, 87
        writing, 36, 94–95, 100
    existence, determining, 65
    naming, 19, 30
        inquiring, 33, 76
        renaming, 32, 93
    number of
        inquiring, 25, 79
    numbering, 14
        inquiring, 31, 77
    scopes
        changing, 94
        constants, 18
            GLOBAL_SCOPE, 18
            VARIABLE_SCOPE, 18
        inquiring, 33, 78

CDF library
    caching scheme
        selecting, 65, 67, 70, 105, 107, 109, 111
    copyright notice
        length, 19
        reading, 81
    interfaces, 21
    modes
        -0.0 to 0.0
            confirming, 65
            constants, 19
                NEGtoPOSfp0off, 19
                NEGtoPOSfp0on, 19
            selecting, 19, 105
        decoding
            confirming, 65
            constants, 16
                ALPHAOSF1_DECODING, 16
                ALPHAVMSd_DECODING, 16
                ALPHAVMSg_DECODING, 16
                DECSTATION_DECODING, 16
                HOST_DECODING, 16
                HP_DECODING, 17
                IBMPC_DECODING, 17
                IBMRS_DECODING, 16
                MAC_DECODING, 17
                NETWORK_DECODING, 16
                NeXT_DECODING, 17
                SGi_DECODING, 16
                SUN_DECODING, 16
                VAX_DECODING, 16
            selecting, 105
        read-only
            confirming, 65
            constants, 18
                READONLYoff, 18
                READONLYon, 18
            selecting, 18, 105
        zMode
            confirming, 66
            constants, 18
                zMODEoff, 18
                zMODEon1, 18
                zMODEon2, 18

selecting, 18, 106
shareable, 9
version
inquiring, 81–82
CDF$INC, 1
CDF$LIB, 5
cdf.inc, 1, 13
CDF_attr_create, 30
CDF_attr_entry_inquire, 35
CDF_attr_get, 38
CDF_attr_inquire, 33
CDF_ATTR_NAME_LEN, 19
CDF_attr_num, 31
CDF_attr_put, 36
CDF_attr_rename, 32
CDF_close, 27
CDF_COPYRIGHT_LEN, 19
CDF_create, 21
CDF_delete, 28
CDF_doc, 24
CDF_error, 29, 120, 125
CDF_INC, 1
CDF_inquire, 25
CDF_LIB, 5
CDF_lib, 55
CDF_lib_x, 63
CDF_MAX_DIMS, 19
CDF_OK, 14
CDF_open, 23
CDF_PATHNAME_LEN, 19
CDF_STATUSTEXT_LEN, 19
CDF_var_close, 53
CDF_var_create, 40
CDF_var_get, 47
CDF_var_hyper_get, 51
CDF_var_hyper_put, 49
CDF_var_inquire, 44
CDF_VAR_LEN, 19
CDF_var_num, 42
CDF_var_put, 46
CDF_var_rename, 43
CDF_WARN, 14
CDFerror, 125
cdfmsf.inc, 1
CDFs
accessing, 23, 28
browsing, 18
closing, 22–23, 27, 64
copyright notice, 0, 24
length, 19
reading, 24, 78
corrupted, 21, 28
creating, 21, 72
current, 58
confirming, 65
selecting, 105
deleting, 21, 28, 74
encoding
changing, 94
constants, 15
ALPHAOSF1_ENCODING, 16
ALPHAVMSd_ENCODING, 15
ALPHAVMSg_ENCODING, 15
DECSTATION_ENCODING, 15
HOST_ENCODING, 15
HP_ENCODING, 16
IBMPC_ENCODING, 16
IBMRS_ENCODING, 16
MAC_ENCODING, 16
NETWORK_ENCODING, 15
NeXT_ENCODING, 16
SGi_ENCODING, 15
SUN_ENCODING, 15
VAX_ENCODING, 15
default, 73
inquiring, 25, 78
format
changing, 94
constants, 14
MULTI_FILE, 14
SINGLE_FILE, 14
default, 73
inquiring, 78
naming, 19, 21, 23
opening, 23, 93
overwriting, 21
version, 24
inquiring, 24, 78–80
COL_MAJOR, 17
compiling
Fortran applications, 1
compute_EPOCH, 121

data types
constants, 14
CDF_BYTE...CDF_EPOCH, 14–15
EPOCH, 121
inquiring size, 80
DECSTATION_DECODING, 16
DECSTATION_ENCODING, 15
definitions file, 1, 5
DEFINITIONS.COM, 1, 5

definitions.csh, 1, 5
definitions.ksh, 1, 5
definitions.sh, 1, 5
definitions.tcsh, 1, 5
dimensions
  limit, 19
  numbering, 14

encode_EPOCH, 122
encode_EPOCH1, 122
encode_EPOCH2, 123
encode_EPOCH3, 123
EPOCH
  breaking down, 121
  computing, 121
  encoding, 122–123
  parsing, 122
  utility routines, 121
    Fortran applications
      compute_EPOCH, 121
      encode_EPOCH, 122
      encode_EPOCH1, 122
      encode_EPOCH2, 123
      encode_EPOCH3, 123
      EPOCH_breakdown, 121
      parse_EPOCH, 122
EPOCH_breakdown, 121
examples
  Fortran applications
    closing
      CDF, 27
      rVariable, 54
    creating
      attribute, 30
      CDF, 22, 55
      rVariable, 41, 112
      zVariable, 112
    deleting
      CDF, 28
    inquiring
      attribute, 34
        entry, 36
        number, 32
      CDF, 24, 26
      explanation text, 29
      rVariable, 45
        number, 42
    Internal Interface, 55, 112
    interpreting
      status codes, 119
    opening
      CDF, 23
    reading
      attribute entry, 39
      rVariable values
        hyper, 52, 113
        single, 48
      zVariable values
        sequential, 115
    renaming
      attribute, 33, 114
      rVariable, 44
    status handler, 119
    writing
      attribute
        gEntry, 37
        rEntry, 37, 116
      rVariable values
        hyper, 50
        single, 46
      zVariable values
        multiple variable, 116

Fortran programming interface, 13
  argument passing, 13
  compiling, 1
  linking, 5
  summary, 133

GLOBAL_SCOPE, 18

HOST_DECODING, 16
HOST_ENCODING, 15
HP_DECODING, 17
HP_ENCODING, 16

IBMPC_DECODING, 17
IBMPC_ENCODING, 16
IBMRS_DECODING, 16
IBMRS_ENCODING, 16
include files, 1
interfaces
  Internal, 55
  mixing, 21
  Standard, 21
Internal Interface, 55
  current objects/states, 58
    attribute, 58
    attribute entries, 58–59
    CDF, 58
    preselected, required, 64
    records/dimensions, 59–61

sequential value, 60–61
status code, 61
variables, 58
examples, 55, 112
indentation/style, 62
items, 64
operations, 64
status codes, returned, 61
syntax, 62
argument list, 62
limitations, 62
item referencing, 14

`libcdf.o`, 7
limits, 19
attribute names, 19
copyright text, 19
dimensions, 19
explanation text, 19
IBM PC, 19
Internal Interface, 62
pathnames, 19
variable names, 19
linking
Fortran applications, 5

`MAC_DECODING`, 17
`MAC_ENCODING`, 16
memory models (MS-DOS), 3, 7
`MULTI_FILE`, 14

`NEGtoPOSfp0off`, 19
`NEGtoPOSfp0on`, 19
`NETWORK_DECODING`, 16
`NETWORK_ENCODING`, 15
`NeXT_DECODING`, 17
`NeXT_ENCODING`, 16
`NOVARY`, 17

`parse_EPOCH`, 122

`READONLYoff`, 18
`READONLYon`, 18
`ROW_MAJOR`, 17

`SGi_DECODING`, 16
`SGi_ENCODING`, 15
`SINGLE_FILE`, 14
Standard Interface, 21
status codes, 125
constants, 14, 125
`CDF_OK`, 14
`CDF_WARN`, 14
current, 61
selecting, 105
error, 125
explanation text, 125
inquiring, 29, 87, 125
length, 19
informational, 125
interpreting, 119, 125
returned, 21
status handler, example, 119
warning, 125
`SUN_DECODING`, 16
`SUN_ENCODING`, 15

`VARIABLE_SCOPE`, 18
variables
accessing, 21
hyper values
dimension counts
current, 60–61
confirming, 68, 70
selecting, 107, 110
dimension indices, starting
current, 59, 61
confirming, 68, 70
selecting, 108, 110
dimension intervals
current, 60–61
confirming, 68, 71
selecting, 108, 110
reading, 51, 83, 89
record count
current, 59–60
confirming, 69, 71
selecting, 108, 110
record interval
current, 59–60
confirming, 69, 71
selecting, 108, 110
record number, starting
current, 59–60
confirming, 69, 71
selecting, 108, 111
writing, 49, 98, 102
multiple variable
reading, 87, 92
record numbers
current, 59–60
selecting, 111

writing, 99, 104
sequential values
current value, 60–61
confirming, 68, 71
selecting, 107, 111
reading, 86, 92
writing, 99, 103
single values
dimension indices
current, 59, 61
confirming, 68, 70
selecting, 108, 110
reading, 47, 83, 88
record number
current, 59–60
confirming, 69, 71
selecting, 108, 111
writing, 46, 96, 101
closing, 53, 64
creating, 40, 73
current, 58
confirming, 67, 70
selecting
by name, 107, 109
by number, 106, 109
data specification
changing, 97, 101
data type
inquiring, 44, 83, 89
number of elements
inquiring, 44, 85, 91
deleting, 75
dimensionality
inquiring, 25–26, 86, 89, 91
existence, determining, 68, 70
indices
numbering, 14
majority
changing, 94
considering, 17
constants, 17
`COL_MAJOR`, 17
`ROW_MAJOR`, 17
default, 73
inquiring, 26, 78
naming, 19, 40
inquiring, 44, 84, 90
renaming, 43, 98, 103
number of, inquiring, 25, 79
numbering, 14
inquiring, 42, 85, 91
pad values
inquiring, 85, 91
specifying, 98, 103
records
allocated
inquiring, 84, 90
specifying, 96, 101
extend
inquiring, 83, 89
specifying, 97, 102
indexing
inquiring, 84–85, 90–91
initial
writing, 98, 102
maximum
inquiring, 25, 84, 86, 90, 92
numbering, 14
variances
constants, 17
`NOVARY`, 17
`VARY`, 17
dimensional
changing, 97, 101
inquiring, 44, 83, 89
record
changing, 99, 103
inquiring, 44, 86, 92
`VARY`, 17
`VAX_DECODING`, 16
`VAX_ENCODING`, 15

`zMODEoff`, 18
`zMODEon1`, 18
`zMODEon2`, 18