

Perl to CDF Programming Interface- V3.1

Using the Package

CDF's Internal Interface is fully covered by the Perl codes in the Perl-CDF package module. For CDF's Standard Interface, only the original functions are covered in the Standard Interface. Refer to the C Reference Manual to see how to pass the arguments to a operation in the Internal Interface function.

In order to use either one or both from Perl, the search path needs to be set up properly. The Perl-CDF package needs to be imported to the Perl script which will access CDF library. There are two ways to accomplish this. One is to include the needed directories in the script. For examples, the following codes are required at the front section of a Perl script using the Perl-CDF package stored an account at /home/cdf/PerlCDF31:

```
use strict;

BEGIN { unshift @INC, '/home/cdf/PerlCDF31/blib/arch',
                  '/home/cdf/PerlCDF31/blib/lib' ) }

use CDF;
```

The other way is to have both

```
-I/home/cdf/cdf31-dist/PerlCDF31/blib/arch
-I/home/cdf/cdf31-dist/PerlCDF31/blib/lib
```

at the command line while running the Perl script.

The shared version of the CDF library is used while accessing the CDF interface from the Perl. To access the most current shared CDF library, an environment variable may need to be set. On a DEC Alpha/OSF1, Sun Solaris or SGI, the environment variable LD_LIBRARY_PATH must be set to be the directory containing libcdf.so. For example, if the most recent CDF V3.1 is at /usr/local/share/cdf31/lib on xfiles, so under the C-shell, enter:

```
setenv LD_LIBRARY_PATH /usr/local/share/cdf31/lib
```

For BSD-based Mac OS X, the environment variable is DYLD_LIBRARY_PATH. The variable must be set to the directory containing the shared library libcdf.dylib.

For Windows 9x/NT/2000/XP, similarly, set the PATH to include the directory that contains libcdf.lib.

Two Perl test scripts, testPerlCDFii.pl and testPerlCDFsi.pl, are provided in the distribution. Both use extensive Perl-CDF interface functions: testPerlCDFii.pl tests CDF's Internal Interface functions while testPerlCDFsi.pl tests the Standard Interface functions. They can be used as sample scripts for development.

New functions are added to allow creating backward compatible CDF files in an older, i.e., V2.7, version, instead of the default, V3.1. Use functions CDFsetFileBackward and CDFgetFileBackward to set and check

the backward flag. If you want to create CDF files with the default version, then don't use CDFsetFileBackward function. Or, call the function with a value of BACKWARDFILEoff (defined in the CDF package). Calling the function with a value of BACKWARDFILEon will cause the CDF to be created as an older version file. Make sure that the flag is set before a file is created.

Creating Backward Compatible File

```
CDF::CDFsetFileBackward ($flag)
my $flag;                                /* in */

my $flag=CDF::CDFgetFileBackward ()
```

Standard Interface

```
my $status=CDF::CDFcreate ($CDFpath, $numDims, \@dimSizes, $encoding,
                           $majority, \$id)

my $CDFpath;                                /* in */
my $numDims;                                /* in */
my @dimSizes;                                /* in */
my $encoding;                                /* in */
my $majority;                                /* in */
my $id;                                      /* out */

my $status=CDF::CDFopen ($CDFpath, \$id)
my $CDFpath;                                /* in */
my $id;                                      /* out */

my $status= CDF::CDFdoc ($id, \$version, \$release, \$text)
my $id;                                      /* in */
my $version;                                /* out */
my $release;                                /* out */
my $text;                                    /* out */

my $status=CDF::CDFinquire ($id, \$numDims, \@dimSizes, \$encoding,
                            \$majority, \$maxRec, \$numVars, \$numAttrs)

my $id;                                      /* in */
my $numDims;                                /* out */
my @dimSizes;                                /* out */
my $encoding;                                /* out */
my $majority;                                /* out */
my $maxRec;                                  /* out */
my $numVars;                                /* out */
my $numAttrs;                                /* out */

my $status=CDF::CDFclose ($id)
my $id;                                      /* in */
```

```

my $status=CDF::CDFdelete ($id)
my $id; /* in */

my $status=CDF::CDFerror ($statusi, \ $message)
my $statusi; /* in */
my $message; /* out */

my $status=CDF::CDFattrCreate ($id, $attrName, $attrScope, \ $attrNum)
my $id; /* in */
my $attrName; /* in */
my $attrScope; /* in */
my $attrNum; /* out */

my $attrNum=CDF::CDFattrNum ($id, $attrName)
my $id; /* in */
my $attrName; /* in */

my $status=CDF::CDFattrRename ($id, $attrNum, $attrName)
my $id; /* in */
my $attrNum; /* in */
my $attrName; /* in */

my $status=CDF::CDFattrInquire ($id, $attrNum, \ $attrName, \ $attrScope,
                                \ $maxEntry)
my $id; /* in */
my $attrNum; /* in */
my $attrName; /* out */
my $attrScope; /* out */
my $maxEntry; /* out */

my $status=CDF::CDFattrEntryInquire ($id, $attrNum, $entryNum, \ $dataType,
                                     \ $numElements)
my $id; /* in */
my $attrNum; /* in */
my $entryNum; /* in */
my $dataType; /* out */
my $numElements; /* out */

my $status=CDF::CDFattrPut ($id, $attrNum, $entryNum, $dataType, $numElements,
                            \ $value)
my $id; /* in */
my $attrNum; /* in */
my $entryNum; /* in */
my $dataType; /* in */
my $numElements; /* in */
my $value; /* in */

my $status=CDF::CDFattrGet ($id, $attrNum, $entryNum, \ $value)
my $id; /* in */
my $attrNum; /* in */

```

```

my $entryNum; /* in */
my $value; /* out */

my $status=CDF::CDFvarCreate ($id, $varName, $dataType, $numElements,
                             $recVariances, \@dimVariances, \$varNum)

my $id; /* in */
my $varName; /* in */
my $dataType; /* in */
my $numElements; /* in */
my $recVariance; /* in */
my @dimVariances; /* in */
my $varNum; /* out */

my $varNum=CDF::CDFvarNum ($id, $varName)

my $id; /* in */
my $varName; /* in */

my $status=CDF::CDFvarRename ($id, $varNum, $varName)

my $id; /* in */
my $varNum; /* in */
my $varName; /* in */

my $status=CDF::CDFvarInquire ($id, $varNum, $varName, $dataType,
                              $numElements, $recVariance, \@dimVariances)

my $id; /* in */
my $varNum; /* in */
my $varName; /* out */
my $dataType; /* out */
my $numElements; /* out */
my $recVariance; /* out */
my @dimVariances; /* out */

my $status=CDF::CDFvarPut ($id, $varNum, $recNum, \@indices, \$value)

my $id; /* in */
my $varNum; /* in */
my $recNum; /* in */
my @indices; /* in */
my $value; /* in */

my $status=CDF::CDFvarGet ($id, $varNum, $recNum, \@indices, \$value)

my $id; /* in */
my $varNum; /* in */
my $recNum; /* in */
my @indices; /* in */
my $value; /* out */

my $status=CDF::CDFvHpPut ($id, $varNum, $recStart, $recCount,
                          $recInterval, \@indices, \@counts,
                          \@intervals, \@buffer)

my $id; /* in */
my $varNum; /* in */

```

```

my $recStart; /* in */
my $recCount; /* in */
my $recInterval; /* in */
my @indices; /* in */
my @counts; /* in */
my @intervals; /* in */
my @buffer; /* in */

my $status=CDF::CDFvHpGet ($id, $varNum, $recStart, $recCount,
                           $recInterval, \@indices, \@counts,
                           \@intervals, \@buffer)

my $id; /* in */
my $varNum; /* in */
my $recStart; /* in */
my $recCount; /* in */
my $recInterval; /* in */
my @indices; /* in */
my @counts; /* in */
my @intervals; /* in */
my @buffer; /* out */

my $status=CDF::CDFvarClose ($id, $varNum)
my $id; /* in */
my $varNum; /* in */

```

Internal Interface

```

my $status=CDF::CDFlib ($op, ...)
my $op; /* in */

CLOSE_
    CDF_
    rVAR_
    zVAR_

CONFIRM_
    ATTR_ /* out */
    ATTR_EXISTENCE_ $attrName /* in */
    CDF_ $id /* out */
    CDF_ACCESS_
    CDF_CACHESIZE_ $numBuffers /* out */
    CDF_DECODING_ $decoding /* out */
    CDF_NAME_ $CDFpath /* out */
    CDF_NEGtoPOSfp0_MODE_ $mode /* out */
    CDF_READONLY_MODE_ $mode /* out */
    CDF_STATUS_ $status /* out */
    CDF_zMODE_ $mode /* out */
    COMPRESS_CACHESIZE_ $numBuffers /* out */
    CURgENTRY_EXISTENCE_

```

CURrENTRY_EXISTENCE_		
CURzENTRY_EXISTENCE_		
gENTRY_	\\$entryNum	/* out */
gENTRY_EXISTENCE_	\$entryNum	/* in */
rENTRY_	\\$entryNum	/* out */
rENTRY_EXISTENCE_	\$entryNum	/* in */
rVAR_	\\$varNum	/* out */
rVAR_CACHESIZE_	\\$numBuffers	/* out */
rVAR_EXISTENCE_	\$varName	/* in */
rVAR_PADVALUE_		
rVAR_RESERVEPERCENT_	\\$percent	/* out */
rVAR_SEQPOS_	\\$recNum	/* out */
	\@indices	/* out */
rVARs_DIMCOUNTS_	\@counts	/* out */
rVARs_DIMINDICES_	\@indices	/* out */
rVARs_DIMINTERVALS_	\@intervals	/* out */
rVARs_RECCOUNT_	\\$recCount	/* out */
rVARs_RECINTERVAL_	\\$recInterval	/* out */
rVARs_RECNUMBER_	\\$recNum	/* out */
STAGE_CACHESIZE_	\\$numBuffers	/* out */
zENTRY_	\\$entryNum	/* out */
zENTRY_EXISTENCE_	\$entryNum	/* in */
zVAR_	\\$varNum	/* out */
zVAR_CACHESIZE_	\\$numBuffers	/* out */
zVAR_DIMCOUNTS_	\@counts	/* out */
zVAR_DIMINDICES_	\@indices	/* out */
zVAR_DIMINTERVALS_	\@intervals	/* out */
zVAR_EXISTENCE_	\$varName	/* in */
zVAR_PADVALUE_		
zVAR_RECCOUNT_	\\$recCount	/* out */
zVAR_RECINTERVAL_	\\$recInterval	/* out */
zVAR_RECNUMBER_	\\$recNum	/* out */
zVAR_RESERVEPERCENT_	\\$percent	/* out */
zVAR_SEQPOS_	\\$recNum	/* out */
	\@indices	/* out */
CREATE_		
ATTR_	\$attrName	/* in */
	\$scope	/* in */
	\\$attrNum	/* out */
CDF_	\$CDFpath	/* in */
	\$numDims	/* in */
	\@dimSizes	/* in */
	\\$id	/* out */
rVAR_	\$varName	/* in */
	\$dataType	/* in */
	\$numElements	/* in */
	\$recVary	/* in */
	\$dimVarys	/* in */

	\\$varNum	/* out */
zVAR_	\$varName	/* in */
	\$dataType	/* in */
	\$numElements	/* in */
	\$numDims	/* in */
	\@dimSizes	/* in */
	\$recVary	/* in */
	\$dimVarys	/* in */
	\\$varNum	/* out */
DELETE_		
ATTR_		
CDF_		
gENTRY_		
rENTRY_		
rVAR_		
rVAR_RECORDS_	\$firstRecord	/* in */
	\$lastRecord	/* in */
zENTRY_		
zVAR_		
zVAR_RECORDS_	\$firstRecord	/* in */
	\$lastRecord	/* in */
GET_		
ATTR_MAXgENTRY_	\\$maxEntry	/* out */
ATTR_MAXrENTRY_	\\$maxEntry	/* out */
ATTR_MAXzENTRY_	\\$maxEntry	/* out */
ATTR_NAME_	\\$attrName	/* out */
ATTR_NUMBER_	\$attrName	/* in */
	\\$attrNum	/* out */
ATTR_NUMgENTRIES_	\\$numEntries	/* out */
ATTR_NUMrENTRIES_	\\$numEntries	/* out */
ATTR_NUMzENTRIES_	\\$numEntries	/* out */
ATTR_SCOPE_	\\$scope	/* out */
CDF_COMPRESSION_	\\$cType	/* out */
	\@cParms	/* out */
	\\$cPct	/* out */
CDF_COPYRIGHT_	\\$copyRight	/* out */
CDF_ENCODING_	\\$encoding	/* out */
CDF_FORMAT_	\\$format	/* out */
CDF_INCREMENT_	\\$increment	/* out */
CDF_INFO_	\$path	/* in */
	\\$cType	/* out */
	\@cParms	/* out */
	\\$cSize	/* out */
	\\$uSize	/* out */
CDF_MAJORITY_	\\$majority	/* out */
CDF_NUMATTRS_	\\$numAttrs	/* out */
CDF_NUMgATTRS_	\\$numAttrs	/* out */
CDF_NUMrVARS_	\\$numVars	/* out */

CDF_NUMvATTRS_	\\$numAttrs	/* out */
CDF_NUMzVARS_	\\$numVars	/* out */
CDF_RELEASE_	\\$release	/* out */
CDF_VERSION_	\\$version	/* out */
DATATYPE_SIZE_	\$dataType	/* in */
	\\$numBytes	/* out */
gENTRY_DATA_	\\$value	/* out */
gENTRY_DATATYPE_	\\$dataType	/* out */
gENTRY_NUMELEMS_	\\$numElements	/* out */
LIB_COPYRIGHT_	\\$copyRight	/* out */
LIB_INCREMENT_	\\$increment	/* out */
LIB_RELEASE_	\\$release	/* out */
LIB_subINCREMENT_	\\$subincrement	/* out */
LIB_VERSION_	\\$version	/* out */
rENTRY_DATA_	\\$value	/* out */
rENTRY_DATATYPE_	\\$dataType	/* out */
rENTRY_NUMELEMS_	\\$numElements	/* out */
rVAR_ALLOCATEDFROM_	\$startRecord	/* in */
	\\$nextRecord	/* out */
rVAR_ALLOCATEDTO_	\$startRecord	/* in */
	\\$lastRecord	/* out */
rVAR_BLOCKINGFACTOR_	\\$blockingFactor	/* out */
rVAR_COMPRESSION_	\\$cType	/* out */
	\@cParms	/* out */
	\\$cPct	/* out */
rVAR_DATA_	\\$value	/* out */
rVAR_DATATYPE_	\\$dataType	/* out */
rVAR_DIMVARYS_	\@dimVarys	/* out */
rVAR_HYPERDATA_	\@buffer	/* out */
rVAR_MAXallocREC_	\\$maxRec	/* out */
rVAR_MAXREC_	\\$maxRec	/* out */
rVAR_NAME_	\\$varName	/* out */
rVAR_nINDEXENTRIES_	\\$numEntries	/* out */
rVAR_nINDEXLEVELS_	\\$numLevels	/* out */
rVAR_nINDEXRECORDS_	\\$numRecords	/* out */
rVAR_NUMallocRECS_	\\$numRecords	/* out */
rVAR_NUMBER_	\$varName	/* in */
	\\$varNum	/* out */
rVAR_NUMELEMS_	\\$numElements	/* out */
rVAR_NUMRECS_	\\$numRecords	/* out */
rVAR_PADVALUE_	\\$value	/* out */
rVAR_RECVAR_	\\$recVary	/* out */
rVAR_SEQDATA_	\\$value	/* out */
rVAR_SPARSEARRAYS_	\\$sArraysType	/* out */
	\@sArraysParms	/* out */
	\\$sArraysPct	/* out */
rVAR_SPARSERECORDS_	\\$sRecordsType	/* out */
rVARs_DIMSIZES_	\@dimSizes	/* out */
rVARs_MAXREC_	\\$maxRec	/* out */
rVARs_NUMDIMS_	\\$numDims	/* out */
rVARs_RECDATA_	\$numVars	/* in */

	\@varNums	/* in */
	\@buffer	/* out */
STATUS_TEXT_	\\$text	/* out */
zENTRY_DATA_	\\$value	/* out */
zENTRY_DATATYPE_	\\$dataType	/* out */
zENTRY_NUMELEMS_	\\$numElements	/* out */
zVAR_ALLOCATEDFROM_	\$startRecord	/* in */
	\\$nextRecord	/* out */
zVAR_ALLOCATEDTO_	\$startRecord	/* in */
	\\$lastRecord	/* out */
zVAR_BLOCKINGFACTOR_	\\$blockingFactor	/* out */
zVAR_COMPRESSION_	\\$cType	/* out */
	\@cParms	/* out */
	\\$cPct	/* out */
zVAR_DATA_	\\$value	/* out */
zVAR_DATATYPE_	\\$dataType	/* out */
zVAR_DIMSIZES_	\@dimSizes	/* out */
zVAR_DIMVARYS_	\@dimVarys	/* out */
zVAR_HYPERDATA_	\@buffer	/* out */
zVAR_MAXallocREC_	\\$maxRec	/* out */
zVAR_MAXREC_	\\$maxRec	/* out */
zVAR_NAME_	\\$varName	/* out */
zVAR_nINDEXENTRIES_	\\$numEntries	/* out */
zVAR_nINDEXLEVELS_	\\$numLevels	/* out */
zVAR_nINDEXRECORDS_	\\$numRecords	/* out */
zVAR_NUMallocRECS_	\\$numRecords	/* out */
zVAR_NUMBER_	\$varName	/* in */
	\\$varNum	/* out */
zVAR_NUMDIMS_	\\$numDims	/* out */
zVAR_NUMELEMS_	\\$numElements	/* out */
zVAR_NUMRECS_	\\$numRecords	/* out */
zVAR_PADVALUE_	\\$value	/* out */
zVAR_RECvary_	\\$recVary	/* out */
zVAR_SEQDATA_	\\$value	/* out */
zVAR_SPARSEARRAYS_	\\$sArraysType	/* out */
	\@sArraysParms	/* out */
	\\$sArraysPct	/* out */
zVAR_SPARSERECORDS_	\\$sRecordsType	/* out */
zVARs_MAXREC_	\\$maxRec	/* out */
zVARs_RECdata_	\$numVars	/* in */
	\@varNums	/* in */
	\@buffer	/* out */
OPEN_		
CDF_	\$CDFpath	/* in */
	\\$id	/* out */
PUT_		
ATTR_NAME_	\$attrName	/* in */
ATTR_SCOPE_	\$scope	/* in */
CDF_COMPRESSION_	\$cType	/* in */

	\@cParms	/* in */
CDF_ENCODING_	\$encoding	/* in */
CDF_FORMAT_	\$format	/* in */
CDF_MAJORITY_	\$majority	/* in */
gENTRY_DATA_	\$dataType	/* in */
	\$numElements	/* in */
	\\$value	/* in */
gENTRY_DATASPEC_	\$dataType	/* in */
	\$numElements	/* in */
rENTRY_DATA_	\$dataType	/* in */
	\$numElements	/* in */
	\\$value	/* in */
rENTRY_DATASPEC_	\$dataType	/* in */
	\$numElements	/* in */
rVAR_ALLOCATEBLOCK_	\$firstRecord	/* in */
	\$lastRecord	/* in */
rVAR_ALLOCATERECS_	\$numRecords	/* in */
rVAR_BLOCKINGFACTOR_	\$blockingFactor	/* in */
rVAR_COMPRESSION_	\$cType	/* in */
	\@cParms	/* in */
rVAR_DATA_	\\$value	/* in */
rVAR_DATASPEC_	\$dataType	/* in */
	\$numElements	/* in */
rVAR_DIMVARYS_	\@dimVarys	/* in */
rVAR_HYPERDATA_	\@buffer	/* in */
rVAR_INITIALRECS_	\$nRecords	/* in */
rVAR_NAME_	\$varName	/* in */
rVAR_PADVALUE_	\\$value	/* in */
rVAR_RECVAR_	\$recVary	/* in */
rVAR_SEQDATA_	\\$value	/* in */
rVAR_SPARSEARRAYS_	\$sArraysType	/* in */
	\@sArraysParms	/* in */
rVAR_SPARSERECORDS_	\$sRecordsType	/* in */
rVARs_RECDA_	\$numVars	/* in */
	\@varNums	/* in */
	\@buffer	/* in */
zENTRY_DATA_	\$dataType	/* in */
	\$numElements	/* in */
	\\$value	/* in */
zENTRY_DATASPEC_	\$dataType	/* in */
	\$numElements	/* in */
zVAR_ALLOCATEBLOCK_	\$firstRecord	/* in */
	\$lastRecord	/* in */
zVAR_ALLOCATERECS_	\$numRecords	/* in */
zVAR_BLOCKINGFACTOR_	\$blockingFactor	/* in */
zVAR_COMPRESSION_	\$cType	/* in */
	\@cParms	/* in */
zVAR_DATA_	\\$value	/* in */
zVAR_DATASPEC_	\$dataType	/* in */
	\$numElements	/* in */
zVAR_DIMVARYS_	\@dimVarys	/* in */

zVAR_INITIALRECS_	\$nRecords	/* in */
zVAR_HYPERDATA_	\@buffer	/* in */
zVAR_NAME_	\$varName	/* in */
zVAR_PADVALUE_	\\$value	/* in */
zVAR_RECVAR_	\$recVary	/* in */
zVAR_SEQDATA_	\\$value	/* in */
zVAR_SPARSEARRAYS_	\$sArraysType	/* in */
	\@sArraysParms	/* in */
zVAR_SPARSERECORDS_	\$sRecordsType	/* in */
zVARs_RECDATA_	\$numVars	/* in */
	\@varNums	/* in */
	\@buffer	/* in */
SELECT_		
ATTR_	\$attrNum	/* in */
ATTR_NAME_	\$attrName	/* in */
CDF_	\$id	/* in */
CDF_CACHESIZE_	\$numBuffers	/* in */
CDF_DECODING_	\$decoding	/* in */
CDF_NEGtoPOSfp0_MODE_	\$mode	/* in */
CDF_READONLY_MODE_	\$mode	/* in */
CDF_SCRATCHDIR_	\$dirPath	/* in */
CDF_STATUS_	\$status	/* in */
CDF_zMODE_	\$mode	/* in */
COMPRESS_CACHESIZE_	\$numBuffers	/* in */
gENTRY_	\$entryNum	/* in */
rENTRY_	\$entryNum	/* in */
rENTRY_NAME_	\$varName	/* in */
rVAR_	\$varNum	/* in */
rVAR_CACHESIZE_	\$numBuffers	/* in */
rVAR_NAME_	\$varName	/* in */
rVAR_RESERVEPERCENT_	\$percent	/* in */
rVAR_SEQPOS_	\$recNum	/* in */
	\@indices	/* in */
rVARs_CACHESIZE_	\$numBuffers	/* in */
rVARs_DIMCOUNTS_	\@counts	/* in */
rVARs_DIMINDICES_	\@indices	/* in */
rVARs_DIMINTERVALS_	\@intervals	/* in */
rVARs_RECCOUNT_	\$recCount	/* in */
rVARs_RECINTERVAL_	\$recInterval	/* in */
rVARs_RECNUMBER_	\$recNum	/* in */
STAGE_CACHESIZE_	\$numBuffers	/* in */
zENTRY_	\$entryNum	/* in */
zENTRY_NAME_	\$varName	/* in */
zVAR_	\$varNum	/* in */
zVAR_CACHESIZE_	\$numBuffers	/* in */
zVAR_DIMCOUNTS_	\@counts	/* in */
zVAR_DIMINDICES_	\@indices	/* in */
zVAR_DIMINTERVALS_	\@intervals	/* in */
zVAR_NAME_	\$varName	/* in */
zVAR_RECCOUNT_	\$recCount	/* in */

zVAR_RECINTERVAL_	\$recInterval	/* in */
zVAR_RECNUMBER_	\$recNum	/* in */
zVAR_RESERVEPERCENT_	\$percent	/* in */
zVAR_SEQPOS_	\$recNum	/* in */
	\@indices	/* in */
zVARs_CACHESIZE_	\$numBuffers	/* in */
zVARs_RECNUMBER_	\$recNum	/* in */

EPOCH Utility Routines

```

my $epoch=CDF::computeEPOCH ($year, $month, $day, $hour, $minute, $second,
                               $msec)

my $year;                                /* in */
my $month;                               /* in */
my $day;                                 /* in */
my $hour;                                /* in */
my $minute;                              /* in */
my $second;                              /* in */
my $msec;                                /* in */

CDF::EPOCHbreakdown ($epoch, \ $year, \ $month, \ $day, \ $hour, \ $minute,
                      \ $second, \ $msec)

my $epoch;                                /* in */
my $year;                                /* out */
my $month;                               /* out */
my $day;                                 /* out */
my $hour;                                /* out */
my $minute;                              /* out */
my $second;                              /* out */
my $msec;                                /* out */

CDF::encodeEPOCH ($epoch, \ $epString)

my $epoch;                                /* in */
my $epString;                             /* out */

CDF::encodeEPOCH1 ($epoch, \ $epString)

my $epoch;                                /* in */
my $epString;                             /* out */

CDF::encodeEPOCH2 ($epoch, \ $epString)

my $epoch;                                /* in */
my $epString;                             /* out */

CDF::encodeEPOCH3 ($epoch, \ $epString)

my $epoch;                                /* in */
my $epString;                             /* out */

CDF::encodeEPOCHx ($epoch, $format, \ $epString)

my $epoch;                                /* in */

```

```

my $format; /* in */
my $epString; /* out */

my $epoch=CDF::parseEPOCH ($epString)
my $epString; /* in */

my $epoch=CDF::parseEPOCH1 ($epString)
my $epString; /* in */

my $epoch=CDF::parseEPOCH2 ($epString)
my $epString; /* in */

my $epoch=CDF::parseEPOCH3 ($epString)
my $epString; /* in */

```

EPOCH16 Utility Routines

```

my $status=CDF::computeEPOCH16 ($year, $month, $day, $hour, $minute, $second,
                                $msec, $usec, $nsec, $psec, \@epoch16)

my $year; /* in */
my $month; /* in */
my $day; /* in */
my $hour; /* in */
my $minute; /* in */
my $second; /* in */
my $msec; /* in */
my $usec; /* in */
my $nsec; /* in */
my $psec; /* in */
my @epoch16; /* out */

CDF::EPOCH16breakdown (\@epoch16, \$year, \$month, \$day, \$hour, \$minute,
                      \$second, \$msec, \$usec, \$nsec, \$psec)

my @epoch16; /* in */
my $year; /* out */
my $month; /* out */
my $day; /* out */
my $hour; /* out */
my $minute; /* out */
my $second; /* out */
my $msec; /* out */
my $usec; /* out */
my $nsec; /* out */
my $psec; /* out */

CDF::encodeEPOCH16 (\@epoch16, \$epString)
my @epoch16; /* in */
my $epString; /* out */

```

```

CDF::encodeEPOCH16_1 (\@epoch16, \$sepString)
my @epoch16; /* in */
my $sepString; /* out */

CDF::encodeEPOCH16_2 (\@epoch16, \$sepString)
my @epoch16; /* in */
my $sepString; /* out */

CDF::encodeEPOCH16_3 (\@epoch16, \$sepString)
my @epoch16; /* in */
my $sepString; /* out */

CDF::encodeEPOCH16_x (\@epoch16, $format, \$sepString)
my @epoch16; /* in */
my $format; /* in */
my $sepString; /* out */

my $status=CDF::parseEPOCH16 ($sepString, \@epoch16)
my $sepString; /* in */
my @epoch16; /* out */

my $status=CDF::parseEPOCH16_1 ($sepString, \@epoch16)
my $sepString; /* in */
my @epoch16; /* out */

my $status=CDF::parseEPOCH16_2 ($sepString, \@epoch16)
my $sepString; /* in */
my @epoch16; /* out */

my $status=CDF::parseEPOCH16_3 ($sepString, \@epoch16)
my $sepString; /* in */
my @epoch16; /* out */

```

Summary

C Interface (Input)	Perl Interface
-----	-----
CDFid varname;	\$varname
char varname;	\$varname
char *varname;	\$varname
char varname[];	\$varname
double varname;	\$varname
long varname;	\$varname
long varname[];	\@varname
void *buffer; (>= 1 value)	\@buffer
void *value; (1 value)	\\$value
double epoch16[];	\@epoch16

C Interface (Output)	Perl Interface
-----	-----
CDFid *varname;	\\\$varname
char *varname;	\\\$varname
char varname[];	\\\$varname
long *varname;	\\\$varname
long varname[];	\\@varname
void *buffer; (>= 1 value)	\\@buffer
void *value; (1 value)	\\\$value
double *epoch16	\\@epoch16