The pad value for a variable may be specified with the Internal Interface. It should be specified before any values are read from or written to the variable; otherwise the default pad value will be used. The pad value may be changed at any time (and any number of times) and will be in effect for all subsequent operations.

The default pad value for each data type is as follows:[20]

```
CDF_BYTE        0
CDF_INT1        0
CDF_UINT1       0
CDF_INT2        0
CDF_UINT2       0
CDF_INT4        0
CDF_UINT4       0
CDF_REAL4       0.0
CDF_FLOAT       0.0
CDF_REAL8       0.0
CDF_DOUBLE      0.0
CDF_EPOCH       01-Jan-0000 00:00:00.000 (0.0)
CDF_CHAR        " " (space character, 40 octal)
CDF_UCHAR       " " (space character, 40 octal)
```

2.4 Attributes

CDF attributes are the mechanism for storing metadata. (Variables are used to store data.) A new attribute may be created in a CDF at any time.

2.4.1 Naming

Each attribute in a CDF has a unique name. Attribute names are case sensitive regardless of the operating system being used and may consist of up to `CDF_ATTR_NAME_LEN` printable characters (including blanks). Trailing blanks, however, are ignored when the CDF library compares attribute names. `"UNITS"` and `"UNITS   "` are considered to be the same name, so they cannot both exist in the same CDF. This was done because Version 1 of CDF padded attribute names on the right with blanks out to eight characters. When a Version 1 CDF was converted to a Version 2 CDF these trailing blanks remained in the attributes names. To allow CDF Version 2 applications to read such a CDF without having to be concerned with the trailing blanks, the trailing blanks are ignored by the CDF when comparing attributes names. The trailing blanks are returned as part of the name, however, when an attribute is inquired by an application program.

2.4.2 Numbering

The attributes in a CDF are numbered consecutively starting at one (1) for Fortran applications and starting at zero (0) for C applications. The CDF library assigns attribute numbers as the attributes are created. Note that there are not separate lists of global and variable scoped attributes. Only one list of attributes exists in a CDF (containing both global and variable scoped attributes).

[20] These default pad values can be changed by your system manager when the CDF distribution is built.

2.4.3 Attribute Scopes

Attribute scopes declare the intended purpose of an attribute. Global scope attributes (gAttributes) describe some aspect of the entire CDF. Variable scope attributes (vAttributes) describe some property of each variable.

An attribute's scope exists to assist in the interpretation of its entries by CDF toolkit programs and user applications (e.g., entries of a vAttribute should correspond to variables). The CDF library also places some restrictions on the operations that may be performed on an attribute of a particular scope.[21] These restrictions consist of the following:

1. A gEntry operation may not be performed on a vAttribute.
2. A zEntry or rEntry operation may not be performed on a gAttribute.
3. While in zMode, only zEntry operations may be performed on vAttributes (see Section 2.1.2).

All other operations involving attributes and their entries remain available.

Assumed Scopes

CDF Version 1 did not allow the scope of an attribute to be explicitly declared. This led to ambiguities in the interpretation of attribute entries in the toolkit programs and user applications. CDF Version 2 does allow the scope of an attribute to be declared when the attribute is created. To ease the transition from Version 1 to Version 2, CDF distributions prior to CDF V2.5 contained the notion of assumed attribute scopes. Assumed attribute scopes arose when the CDF library had to guess the scope of an attribute in a Version 1 CDF (e.g., when the `CDFconvert` program converted a Version 1 CDF to a Version 2 CDF). Beginning with CDF V2.5, all assumed attribute scopes are converted to the corresponding definite scope. When a CDF is read this conversion occurs only in the CDF library — the CDF is not physically altered. When an existing CDF is written to, each assumed attribute scope detected will be physically converted to the corresponding definite scope. Note that if this automatic conversion is incorrect, the scope of an attribute can be corrected using the Internal Interface in an application program or by editing the CDF with the `CDFedit` program.

2.4.4 Deleting

An attribute may be deleted from a CDF. Deleting an attribute also deletes the corresponding entries. The disk space used by the attribute definition and the corresponding entries becomes available for use as needed by the CDF library. Also, the attributes which numerically follow the attribute being deleted are renumbered immediately. (Each is decremented by one.) Attributes are deleted using the `<DELETE_,ATTR_>` operation of the Internal Interface.

2.4.5 Attribute Entries

Attribute entries are used to actually store metadata. Each attribute in a CDF may have zero or more associated entries. For vAttributes two types of entries are supported: rEntries and zEntries. rEntries describe some property of the corresponding rVariable, and zEntries describe some property of the corresponding

[21] This was not necessarily the case in previous releases of CDF. These new restrictions should not, however, cause any conflicts with existing applications.

zVariable. Note that an entry does not have to exist for each variable in the CDF. For gAttributes only one type of entry is supported and is referred to as a gEntry. The gEntries are independent of anything else in the CDF and have meaning only to the application. Note that gEntries are sometimes referred to simply as "entries."

Accessing

The Standard Interface deals exclusively with rEntries (for vAttributes) and gEntries (for gAttributes). No access to zEntries is provided. The Internal Interface may be used to access any type of attribute entry.

Numbering

The rEntries and zEntries for a vAttribute and the gEntries for a gAttribute are numbered starting at one (1) for Fortran applications and starting at zero (0) for C applications. For vAttributes the entry numbers are in fact the variable numbers of the variables being described. rEntries correspond to rVariables and zEntries correspond to zVariables. For gAttributes the gEntry numbers have meaning only to the application.

The entry numbers used need not be contiguous (as are variable and attribute numbers). An application may choose to write any combination of entries for a particular attribute (keeping in mind that the entry numbers used for a vAttribute correspond to the existing variables).

Data Specification

Each entry for an attribute has a data specification and an associated value. A data specification consists of a data type and a number of elements of that data type. The supported data types are described in Section 2.2.10. The entries for an attribute may have any combination of data specifications.

For character string data types the number of elements is the number of characters in the string. For example, if a gEntry value for a gAttribute named `TITLE` were `"Example CDF Title."` (not including the double quotes), the data type would be `CDF_CHAR`, and the number of elements would be 18 (a character string of size 18).

For non-character string data types the number of elements is the size of an array of the data type. For example, if a zEntry value of a vAttribute named `RANGE` were `[100.0,900.0]`, the data type would be `CDF_REAL4`, and the number of elements would be two (an array of two values).

Deleting

An entry may be deleted from an attribute. The disk space used by the entry becomes available for use as needed by the CDF library. There is no renumbering of entries (as with deleting a variable or attribute). Entries are deleted using the `<DELETE_,gENTRY_>`, `<DELETE_,rENTRY_>`, and `<DELETE_,zENTRY_>` operations of the Internal Interface.

2.5 NSSDC Standard

CDFs generated for use by NSSDC applications (e.g., the NSSDC Graphics System [NGS]) are expected to conform to a specified standard. These applications expect certain attributes and make use of several optional attributes and variables to provide additional features. The NSSDC standard does not deal with zVariables or zEntries (for vAttributes).

2.5.1 Standard Attributes

There are several attributes that are required to exist in a CDF if it is to conform to the NSSDC standard. These required attributes are shown in Table 2.6. Several optional attributes are also shown.

	Name	Data Type	Number of Elements[a]
Required gAttributes:			
	`TITLE`	Character (CDF_CHAR)	50
Required vAttributes:			
	`FIELDNAM`	Character (CDF_CHAR)	20
	`VALIDMIN`	(same as variable)	(same as variable)
	`VALIDMAX`	(same as variable)	(same as variable)
	`UNITS`	Character (CDF_CHAR)	20
	`FORMAT`	Character (CDF_CHAR)	8
Optional gAttributes:			
	`TEXT`	Character (CDF_CHAR)	80
	`MODS`	Character (CDF_CHAR)	80
Optional vAttributes:			
	`SCALEMIN`	(same as variable)	(same as variable)
	`SCALEMAX`	(same as variable)	(same as variable)
	`MONOTON`[b]	Character (CDF_CHAR)	8
	`CATDESC`	Character (CDF_CHAR)	40
	`FILLVAL`	(same as variable)	(same as variable)

[a] If the data type is `CDF_CHAR`, blank pad the value on the right to the required number of elements (characters).
[b] We strongly suggest using the `MONOTON` attribute.

Table 2.6: NSSDC Standard CDF Attributes

The purpose of each attribute is defined as follows...

`TITLE` A title for the CDF data set. Only one gEntry should be written (gEntry number zero [0] for C applications, gEntry number one [1] for Fortran applications).

`FIELDNAM` The field name for an rVariable. One rEntry per rVariable should be written.

`VALIDMIN` The minimum valid value for an rVariable. Values below this minimum are treated as invalid or fill data. One rEntry per rVariable should be written.

`VALIDMAX`	The maximum valid value for an rVariable. Values above this maximum are treated as invalid or fill data. One rEntry per rVariable should be written.
`UNITS`	The units of measurement for an rVariable. One rEntry per rVariable should be written.
`FORMAT`	The Fortran format specification to use when displaying the values for an rVariable. One rEntry per rVariable should be written.
`TEXT`	Documentation text about the CDF. Any number of gEntries may be written.
`MODS`	A modification history for the CDF. Each time the CDF is modified, a new gEntry would be added with an explanation of the changes.
`SCALEMIN`	The scale minimum for an rVariable. This is used when scaling the rVariable's values for display. If omitted, the `VALIDMIN` vAttribute rEntry will be used. One rEntry per rVariable should be written.
`SCALEMAX`	The scale maximum for an rVariable. This is used when scaling the rVariable's values for display. If omitted, the `VALIDMAX` vAttribute rEntry will be used. One rEntry per rVariable should be written.
`MONOTON`	The monotonicity of an rVariable. This specifies if the values strictly increase (`"INCREASE"`), strictly decrease (`"DECREASE"`), or are not monotonic (`"FALSE   "`) along a dimension or from record to record. Specifying an rVariable's monotonicity can greatly improve the efficiency of certain NSSDC applications (namely, the NSSDC Graphics System). One rEntry per rVariable should be written.
`CATDESC`	A catalog description for an rVariable. This is used for cataloging purposes by some NSSDC applications. One rEntry per rVariable should be written.
`FILLVAL`	The fill value for an rVariable. This value should not be confused with the pad value that may be specified for an rVariable (with the Internal Interface) that the CDF library actually uses when writing records. The values should agree but the `FILLVAL` vAttribute is not used by the CDF library in any way. It is, however, used by CDF applications (including the CDF toolkit). One rEntry per rVariable should be written.

2.5.2 Standard Variables

There are no required CDF rVariables. The first eight characters of rVariable names must, however, be unique for NSSDC applications. Names longer than eight characters will be truncated when displayed.

The optional rVariable `EPOCH` is used by many NSSDC applications to store date/time values. Section 2.2.10 defines an EPOCH value. The record variance should be `VARY`, each dimension variance should be `NOVARY`, the data type should be `CDF_EPOCH`, and the number of elements should be one (1). If a CDF contains an `EPOCH` variable, the following variable names should not be used: `TIME`, `YEAR`, `MONTH`, `DAY`, `HOUR`, `MINUTE`, `SECOND`, `MSEC`, `IYEAR`, `IMONTH`, `IDAY`, `IHOUR`, `IMINUTE`, `ISECOND`, and `IMSEC`.

Chapter 3

Toolkit Reference

3.1 Introduction

The CDF toolkit is a set of utility programs that allow the creation, analysis, and modification of CDFs. The following sections will describe the use of these programs.

3.1.1 VMS, UNIX, & MS-DOS

Each program is executed at the command line (or may be executed from within your applications using the methods provided by the operating system being used). The following rules apply to the command line syntax:

1. Parameters are required unless noted otherwise. Parameters are shown in angle brackets (<>'s) in the sections which describe each toolkit program.

2. Qualifiers are optional unless noted otherwise.

3. Qualifiers can be truncated as long as no ambiguities result.

4. Optional parts of a command are shown in brackets ([]'s) in the sections which describe each toolkit program.

5. A vertical line (|) is used to separate two or more options in those cases when only one of the options may be specified.

6. Wildcard characters are allowed in CDF names to allow more than one CDF to be specified (where appropriate). Wildcard characters may be used in the CDF name but not the directory path portion of a specification. The wildcard characters supported are similar to those available on the operating system being used.

   UNIX: If a CDF specification is to contain a wildcard character, the entire specification must be enclosed in single quote marks (e.g., `'/disk3/sst*'`).

7. On VMS/OpenVMS systems, qualifiers begin with a slash (/). On UNIX and MS-DOS systems, qualifiers begin with a hyphen (-).

   NOTE: You can override the default notation by specifying a slash or hyphen as the first parameter/qualifier immediately after the program name. When this is done, you may have to adjust the syntax used as follows:

   (a) When the slash notation is used on UNIX systems, character stuffing will be necessary in pathnames (e.g., specify "`//disk1//CDFs`" rather than "`/dist1/CDFs`"). Also, double quote marks are required around options enclosed in parenthesis.

   (b) When the slash notation is used on MS-DOS systems, double quote marks may be needed around entire qualifier/option combinations.

8. On MS-DOS systems the executable names may be different from the names shown in this chapter (file names are limited to 8.3 characters). Where the names differ, the actual name will be noted.

   If you add the directory containing the toolkit executables to your path, you will have to use the 8-character (or fewer) names. If you use a command aliasing program, you could specify the aliases to be the names shown in this chapter with each pointing to the corresponding executable file name.

9. On UNIX systems all parameters/qualifiers entered at the command line are case sensitive. On VMS, OpenVMS, and MS-DOS systems parameters/qualifiers are not case sensitive. Note that variable names are always case sensitive regardless of the operating system being used.

10. If an option contains blanks, it will generally be necessary to enclose the entire option in double quote marks.

11. On UNIX systems, it may be necessary to execute "`stty tab3`" before running `CDFedit`, `CDFlist`, or `CDFwalk`.

In the following sections the available qualifiers and options for each of the toolkit programs will be presented. The default settings for these qualifiers and options will not be shown since they can be configured for a particular CDF distribution. Use `CDFinquire` to determine these defaults.

On VMS/OpenVMS systems you should have executed the command procedure named `DEFINITIONS.COM` before running any of the CDF toolkit programs. This will define the necessary logical names and symbols. Your system administrator knows the location of `DEFINITIONS.COM`.

On UNIX system you should have `source`'d the script file named `definitions.<shell>` where `<shell>` is the name of the shell you are using. This will define the necessary environment variables and aliases. Your system administrator knows the location of `definitions.<shell>`.

3.1.2 Macintosh

Each toolkit program is started by double-clicking on the appropriate icon. A dialog box will be displayed in which the parameters and qualifiers needed to execute the program are specified. When the parameters/qualifiers have been selected, clicking on `Enter` causes the initial execution to begin.

For the programs that use a full-screen interface (e.g., `CDFedit`, `CDFlist`, and `CDFwalk`), a "pasteboard" window is opened in which the program displays menus, prompts, etc. When the "pasteboard" window is closed (by exiting the execution), the parameters/qualifiers dialog box is redisplayed. A new set of parameters/qualifiers may be selected and executed or the program may be terminated.

For the programs that simply output to the screen (e.g., `CDFstats`, `CDFcompare`, and `CDFinquire`), a "standard output" window is opened in which the output will be written. When the execution completes, the "apple" and `File` menus are available in the menu bar. Under the `File` menu the following commands are available:

`Execute`	Causes the parameters/qualifiers dialog box to be redisplayed. A new set of parameters/qualifiers may be selected and executed. The output from each execution is appended to the existing output.
`Save`	Saves the current output to a file named `<program-name>.so` where `<program-name>` is the name of the program.
`Save as...`	Saves the current output to the file specified in the standard output file dialog box that will be displayed.
`Clear`	Clears the current output.
`Quit`	Terminates the program.

The vertical scroll bar as well as the `page up` and `page down` keys may be used to scroll through the output. When a large amount of text has been written, a dialog box may be displayed indicating that an output overflow is about to occur. The output may be saved to a file before being cleared (to allow the execution to continue).

The parameters/qualifiers dialog box for each program uses the standard Macintosh controls. Edit fields are used to enter text values (e.g., the pathname of a CDF). Leaving an edit field blank is allowed in some cases (which will be noted). Check boxes are used to enable or disable a qualifier. An `X` in the check box indicates that the qualifier is enabled. Radio buttons are used in groups to allow one of several options to be chosen for a qualifier. Generally, only one of the radio buttons in a group may be selected.

Several types of files are specified to the toolkit programs. These consist of CDFs, skeleton tables, and output files. In a parameters/qualifiers dialog box edit field or a toolkit program's prompt field a file must be specified using a full or partial pathname. Full pathnames consist of a volume name (which is also the corresponding folder name), zero or more folder names, and finally the file name (with or without an extension). These are all separated by semi-colons (`:`'s). Partial pathnames do not start with a volume name and may start with or without a semi-colon. If a partial pathname starts with a semi-colon, one or more folder names will follow, each separated by a semi-colon, followed by the file name. The first folder must exist in the currently selected folder. If a partial pathname starts without a semi-colon, then only a file name should be present and the file is (or will be) located in the currently selected folder. To ease in the selection of files in parameters/qualifiers dialog boxes, the corresponding edit fields are followed by a `Select` button. When clicked on, a standard input/output file dialog will be displayed in which a file may be specified. When that has been done the pathname of the selected file will appear in the edit field.

A directory/wildcard[1] specification is allowed for some of the CDF specifications required by the toolkit programs. This allows more than one of the CDFs in a directory to be selected. If a CDF specification ends with a folder name, then all of the CDFs in that folder will have been specified. A trailing semi-colon is not required (but may be present). The supported wildcard characters are the asterick (`*`) which matches zero or more characters and the question mark (`?`) which matches exactly one character.

In the following sections the available qualifiers and options for each of the toolkit programs will be presented. The default settings for these qualifiers and options will not be shown since they can be configured for a par-

[1] Macintosh folders are equivalent to the directories discussed here.

ticular CDF distribution. When a program is started, the settings shown in the initial parameters/qualifiers dialog box are the default qualifiers for your CDF distribution.

NOTE: You may find it necessary to increase the partition size available to a toolkit program when dealing with very large CDFs. You can do this by editing the "current size" field of the window opened when using the `Get Info` item of the `File` menu (from the Desktop menu bar) on the toolkit executable.

3.2 CDFdir

3.2.1 Introduction

The `CDFdir` utility is used to display a directory listing of a CDF's files.[2] The `.cdf` file is displayed first followed by the rVariable files and then the zVariable files (if either exist in a multi-file CDF) in numerical order.

3.2.2 Executing the CDFdir Program

The command line syntax for `CDFdir` is as follows:

Usage:

VMS:

```
$ CDFDIR <cdf-path>
```

UNIX:

```
% cdfdir <cdf-path>
```

MS-DOS:

```
> cdfdir <cdf-path>
```

Parameter(s):

`<cdf-path>`

The pathname of the CDF for which to display a directory listing (do not specify an extension).

[2] `CDFdir` is not available on a Macintosh.

Example(s):

VMS:

```
$ CDFDIR NCDS$DATA:GISS_WETL_CLIMATOLOGY
$ CDFDIR [-.TEMP]FGGE3B
```

UNIX:

```
% cdfdir ../cac_sst_blended
% cdfdir ~/CDFs/giss_wetl_climatology
```

MS-DOS:

```
> cdfdir ..\cac_sst
> cdfdir c:\cdfs\gisswetl
```

Help is displayed when `CDFdir` is executed without any arguments.

3.2.3 Output from the CDFdir Program

The format of the output from `CDFdir` is that of a directory listing on the operating system being used.

3.3 CDFcompare

3.3.1 Introduction

The CDFcompare program displays the differences between two CDFs. More than one pair of CDFs can be compared. This program would be used to verify changes made to a CDF (comparing it with the saved original) or to verify the conversions performed by `CDFconvert` (see Section 3.4).

3.3.2 Executing the CDFcompare Program

Usage:

VMS:

```
$ CDFCOMPARE [/[NO]LOG] [/[NO]ATTR] [/[NO]VAR] [/[NO]NUMBER]
             [/[NO]ETC] [/[NO]NEG2POSFP0] [/ZMODES=(<mode1>,<mode2>)]
```

```
               [/[NO]LOCATION] [/REPORT=(<types>)] [/CACHE=<size>]
               [/[NO]PAGE] [/[NO]STATISTICS] [/[NO]PERCENT]
               <cdf-spec-1> <cdf-spec-2>
```

UNIX:

```
% cdfcompare [-[no]log] [-[no]attr] [-[no]var] [-[no]number]
             [-[no]etc] [-[no]neg2posfp0] [-zmodes "<mode1>,<mode2>"]
             [-[no]location] [-report "<types>"] [-cache <size>]
             [-[no]page] [-[no]statistics] [-[no]percent]
             <cdf-spec-1> <cdf-spec-2>
```

MS-DOS:[3]

```
> cdfcompare [-[no]log] [-[no]attr] [-[no]var] [-[no]number]
             [-[no]etc] [-[no]neg2posfp0] [-zmodes "<mode1>,<mode2>"]
             [-[no]location] [-report "<types>"] [-cache <size>]
             [-[no]page] [-[no]statistics] [-[no]percent]
             <cdf-spec-1> <cdf-spec-2>
```

Macintosh:

Double-click on the `CDFcompare` icon. When the desired parameters/qualifiers have been selected in the dialog box, click on the `Enter` button to compare the CDFs. Clicking on the `Help` button will display online help. Clicking on the `Quit` button terminates `CDFcompare`.

Parameter(s):

`<cdf-spec-1> <cdf-spec-2>` (VMS, UNIX, & MS-DOS)
`CDF1` and `CDF2` edit fields (Macintosh)

The specifications of the CDFs to be compared. (Do not enter extensions.) These can be either a pathname specifying a single CDF or a directory/wildcard path specifying more than one CDF. Wildcards are allowed in the CDF name but not in the directory path.

If two directory/wildcard paths are specified, all of the CDFs with matching names will be compared. If a CDF pathname and a directory/wildcard path are specified, the CDF specified will be compared with the CDF in the directory/wildcard path having the same name. If two CDF pathnames are specified, the CDFs are compared. (This is the only way to compare two CDFs having different names.)

Macintosh: At the end of each field a button labeled `Select` is present. When selected, a standard input file dialog is displayed from which a single CDF may be selected.

[3] On MS-DOS systems the executable is named `CDFCMP.EXE`.

Qualifier(s):

/[NO]LOG (VMS)
-[no]log (UNIX & MS-DOS)
Log progress check box (Macintosh)

Specifies whether or not messages about the progress of each comparison are displayed.

/[NO]PERCENT (VMS)
-[no]percent (UNIX & MS-DOS)
Display percentages check box (Macintosh)

Specifies whether or not the percentage of a variable's values compared is displayed during the comparison of that variable. Message logging must also be enabled.

/[NO]ATTR (VMS)
-[no]attr (UNIX & MS-DOS)
Compare attributes check box (Macintosh)

Specifies whether or not attributes (and their entries) are to be compared.

/[NO]VAR (VMS)
-[no]var (UNIX & MS-DOS)
Compare variables check box (Macintosh)

Specifies whether or not variables are to be compared. Note that an rVariable will never be compared with a zVariable.

/[NO]NUMBER (VMS)
-[no]number (UNIX & MS-DOS)
Compare numbers check box (Macintosh)

Specifies whether or not numbering differences between attributes with the same names and between variables with the same names are to be displayed.

/[NO]ETC (VMS)
-[no]etc (UNIX & MS-DOS)
Compare etc. check box (Macintosh)

Specifies whether or not differences transparent to an application will be displayed. These would consist of the version/release/increment of the creating CDF library, format, encoding, etc.

/ZMODES=(<mode1>,<mode2>) (VMS)
-zmodes "<mode1>,<mode2>" (UNIX & MS-DOS)
zMode1 and zMode2 radio buttons (Macintosh)

Specifies the zModes that should be used with the CDF(s) being compared. Note that different zModes may be used for the two CDF(s) specifications. The zModes may be one of the following:

0 Indicates that zMode should be disabled.

1 Indicates that zMode/1 should be used. (The dimension variances of rVariables will be preserved.)

2 Indicates that zMode/2 should be used. (The dimensions of rVariables having a variance of NOVARY [false] are removed.)

`/[NO]NEG2POSFP0` (VMS)
`-[no]neg2posfp0` (UNIX & MS-DOS)
`-0.0 to 0.0` check box (Macintosh)

Specifies whether or not `-0.0` is converted to `0.0` by the CDF library when read from a CDF. `-0.0` is an illegal floating point value on VAXes and DEC Alphas running OpenVMS.

`/[NO]PAGE` (VMS)
`-[no]page` (UNIX & MS-DOS)
`Page output` check box (Macintosh)

Specifies whether or not the output is displayed a page at a time.

`/[NO]LOCATION` (VMS)
`-[no]location` (UNIX & MS-DOS)
`Display locations` check box (Macintosh)

Specifies whether or not the locations of variable value differences are displayed. The locations are displayed in the form

`<record-number>:[<index1>,<index2>,...,<indexN>]`

`/REPORT=(<types>)` (VMS)
`-report "<types>"` (UNIX & MS-DOS)
`Report info's/warnings/errors` check boxes (Macintosh)

Specifies the types of return status codes from the CDF library that should be reported/displayed.

VMS, UNIX, & MS-DOS: `<types>` is a comma-separated list of zero or more of the following symbols: `errors`, `warnings`, or `informationals`. Note that these symbols can be truncated (e.g., `e`, `w`, and `i`).

`/CACHE=<size>` (VMS)
`-cache <size>` (UNIX & MS-DOS)
`Cache size` edit field (Macintosh)

Specifies the cache size to be used for the `.cdf` file of both CDFs in a comparison (i.e., the only files if they are single-file CDFs). If omitted, default cache sizes as chosen by the CDF library are used. The cache size is the number of 512-byte buffers to be used.

`/[NO]STATISTICS` (VMS)
`-[no]statistics` (UNIX & MS-DOS)
`Display statistics` check box (Macintosh)

Specifies whether or not caching statistics are displayed at the end of each comparison.

Example(s):

VMS:

```
$ CDFCOMPARE GISS_WETL GISS_WETL1
$ CDFCOMPARE/LOG/NOATTR/NUMBER/REPORT=(ERRORS) GISS_WETL CDF$SMPL:GISS_WETL
$ CDFCOMPARE/NOVAR/NOETC/ZMODES=(1,2) NCDS$SMPL: NCDS$DATA:
```

UNIX:

```
% cdfcompare giss_wetl giss_wetl1
% cdfcompare -log -noattr -number -report "errors" giss_wetl ../giss_wetlx
% cdfcompare -novar -noetc -zmodes "1,2" /user5/CDFs /user6/CDFs
```

MS-DOS:

```
> cdfcompare gisswetl c:\gisswetl
> cdfcompare -log -noattr -number -report "errors" gisswetl ..\..\giswetlx
> cdfcompare -novar -noetc -zmodes "1,2" a:\cdfs c:\cdf\cdfs
```

VMS, UNIX, & MS-DOS: Command line help is displayed when `CDFcompare` is executed without any arguments.

3.3.3 Output from the CDFcompare Program

The output from `CDFcompare` consists of messages indicating the differences found. If message logging is enabled, the progress of each comparison is also displayed.

3.4 CDFconvert

3.4.1 Introduction

The `CDFconvert` program is used to convert various properties of a CDF. In all cases new CDFs are created. (Existing CDFs are not modified.) Any combination of the following properties may be changed when converting a CDF.

1. A CDF created with an earlier release of the CDF library (e.g., CDF V2.4) may be converted to the current library release.
2. The format of the CDF may be changed (see Section 2.2.7).
3. The data encoding of the CDF may be changed (see Section 2.2.8).
4. The variable majority of the CDF may be changed (see Section 2.3.12).

3.4.2 Executing the CDFconvert Program

Usage:

VMS:

```
$ CDFCONVERT [/SKELETON=<skt-cdf-path>] [/[NO]LOG]
             [/[NO]PERCENT] [/REPORT=(<types>)]
             [/CACHE=<size>] [/[NO]PAGE] [/[NO]STATISTICS]
             <src-cdf-spec>
                [/ZMODE=<mode>] [/[NO]NEG2POSFP0]
             <dst-cdf-spec>
                [/SINGLE | /MULTI] [/ROW | /COLUMN] [/[NO]DELETE]
                [/ENCODING=<encoding> | /HOST | /NETWORK]
```

UNIX:

```
% cdfconvert [-skeleton <skt-cdf-path>] [-[no]log]
             [-[no]percent] [-report "<types>"]
             [-cache <size>] [-[no]page] [-[no]statistics]
             <src-cdf-spec>
                [-zmode <mode>] [-[no]neg2posfp0]
             <dst-cdf-spec>
                [-single | -multi] [-row | -column] [-[no]delete]
                [-encoding <encoding> | -host | -network]
```

MS-DOS:[4]

```
> cdfconvert [-skeleton <skt-cdf-path>] [-[no]log]
             [-[no]percent] [-report "<types>"]
             [-cache <size>] [-[no]page] [-[no]statistics]
             <src-cdf-spec>
                [-zmode <mode>] [-[no]neg2posfp0]
             <dst-cdf-spec>
                [-single | -multi] [-row | -column] [-[no]delete]
                [-encoding <encoding> | -host | -network]
```

Macintosh:

Double-click on the `CDFconvert` icon. When the desired parameters/qualifiers have been selected in the dialog box, click on the `Enter` button to convert the CDFs. Clicking on the `Help` button will display online help. Clicking on the `Quit` button terminates `CDFconvert`.

Parameter(s):

`<src-cdf-spec>` (VMS, UNIX, & MS-DOS)
`Source` edit field (Macintosh)

The source CDF(s). This can be either a single CDF pathname or a directory/wildcard path in which case all CDFs that match the specification will be converted. Wildcards are allowed in the CDF name but not in the directory path. In either case do not specify an extension.

[4] On MS-DOS systems the executable is named `CDFCVT.EXE`.

Macintosh: At the end of this field a button labeled `Select` is present. When selected, a standard input file dialog is displayed from which a single CDF may be selected.

`<dst-cdf-spec>` (VMS, UNIX, & MS-DOS)
`Destination` edit field (Macintosh)

The destination of the converted CDF(s). This may be a single CDF pathname only if a single source CDF was specified. If the directory paths are the same, then a different CDF name must be specified. If the source CDF specification is a directory/wildcard path, then this must be a directory path (other than the source directory path). This may also be a directory path if only a single CDF is being converted. In any case do not specify an extension.

Macintosh: At the end of this field a button labeled `Select` is present. When selected, a standard output file dialog is displayed in which a single CDF may be specified.

Qualifier(s):

`/SKELETON=<skt-cdf-path>` (VMS)
`-skeleton <skt-cdf-path>` (UNIX & MS-DOS)
`Skeleton` edit field (Macintosh)

The pathname of a skeleton CDF to be used during the conversions. (Do not enter an extension.) The skeleton CDF is used in the following cases:

1. If a format for the destination CDF was not specified, then the format of the skeleton CDF will be used.
2. If a variable majority for the destination CDF was not specified, then the variable majority of the skeleton CDF will be used.
3. If a data encoding for the destination CDF was not specified, then the data encoding of the skeleton CDF will be used.

Specifying a skeleton CDF is optional.

Macintosh: At the end of this field a button labeled `Select` is present. When selected, a standard input file dialog is displayed from which the skeleton CDF may be selected.

`/[NO]LOG` (VMS)
`-[no]log` (UNIX & MS-DOS)
`Log progress` check box (Macintosh)

Specifies whether or not messages about the progress of each conversion are displayed.

`/[NO]PAGE` (VMS)
`-[no]page` (UNIX & MS-DOS)
`Page output` check box (Macintosh)

Specifies whether or not the output is displayed a page at a time.

`/[NO]PERCENT` (VMS)
`-[no]percent` (UNIX & MS-DOS)
`Display percentages` check box (Macintosh)

Specifies whether or not the percentage of a variable's values converted is displayed during the conversion of that variable. Message logging must also be enabled.

`/[NO]DELETE` (VMS)
`-[no]delete` (UNIX & MS-DOS)
`Delete existing` check box (Macintosh)

Specifies whether or not a destination CDF is deleted if it already exists.

`/SINGLE | /MULTI` (VMS)
`-single | -multi` (UNIX & MS-DOS)
`Source/Single/Multi` radio buttons (Macintosh)

The format of the destination CDF(s).

VMS, UNIX, & MS-DOS: This overrides the format of the skeleton CDF (if one was specified). If neither this qualifier nor a skeleton CDF is specified, then the format of a destination CDF will be the same as that of the source CDF.

Macintosh: Selecting `Single` or `Multi` will override the format of the skeleton CDF (if one was specified). If `Source` is selected (and a skeleton CDF was not specified), then the format of a destination CDF will be the same as that of the source CDF. (Selecting `Source` will not override the format of a specified skeleton CDF.)

`/ROW | /COLUMN` (VMS)
`-row | -column` (UNIX & MS-DOS)
`Source/Row/Column` radio buttons (Macintosh)

The variable majority of the destination CDF(s).

VMS, UNIX, & MS-DOS: This overrides the variable majority of the skeleton CDF (if one was specified). If neither this qualifier nor a skeleton CDF is specified, then the variable majority of a destination CDF will be the same as that of the source CDF.

Macintosh: Selecting `Row` or `Column` will override the variable majority of the skeleton CDF (if one was specified). If `Source` is selected (and a skeleton CDF was not specified), then the variable majority of a destination CDF will be the same as that of the source CDF. (Selecting `Source` will not override the variable majority of a specified skeleton CDF.)

`/ENCODING=<encoding> | /HOST | /NETWORK` (VMS)
`-encoding <encoding> | -host | -network` (UNIX & MS-DOS)
`Source/Host/Network/Sun...Vax` radio buttons (Macintosh)

The data encoding of the destination CDF(s).

VMS, UNIX, & MS-DOS: This overrides the data encoding of the skeleton CDF (if one was specified). If neither this qualifier nor a skeleton CDF is specified, then the data encoding of a destination CDF will be the same as that of the source CDF. The possible values of `<encoding>` are `host`, `network`, `sun`, `vax`, `decstation`, `sgi`, `ibmpc`, `ibmrs`, `mac`, `hp`, `next`, `alphaosf1`, `alphavmsd`, and `alphavmsg` (and their uppercase equivalents). Note that the host and network qualifiers are no longer necessary (but are supported for compatibility with previous CDF distributions).

Macintosh: Selecting `Host`, `Network`, or a specific machine will override the data encoding of the skeleton CDF (if one was specified). If `Source` is selected (and a skeleton CDF was not specified), then the data encoding of a destination CDF will be the same as that of the source CDF. (Selecting `Source` will not override the data encoding of a specified skeleton CDF.)

`/ZMODE=<mode>` (VMS)
`-zmode <mode>` (UNIX & MS-DOS)

`zMode` radio buttons (Macintosh)

Specifies the zMode that should be used with the source CDF(s). The zMode may be one of the following:

- `0` Indicates that zMode should be disabled.
- `1` Indicates that zMode/1 should be used. (The dimension variances of rVariables will be preserved.)
- `2` Indicates that zMode/2 should be used. (The dimensions of rVariables having a variance of `NOVARY` [false] are removed.)

Note that using zMode/1 or zMode/2 on a source CDF that contains rVariables will produce a destination CDF containing only zVariables. The zMode "view" provided for the source CDF is written to the destination CDF during the conversion.

`/[NO]NEG2POSFP0` (VMS)
`-[no]neg2posfp0` (UNIX & MS-DOS)
`-0.0 to 0.0` check box (Macintosh)

Specifies whether or not `-0.0` is converted to `0.0` by the CDF library when read from or written to a CDF. `-0.0` is an illegal floating point value on VAXes and DEC Alphas running OpenVMS.

`/REPORT=(<types>)` (VMS)
`-report "<types>"` (UNIX & MS-DOS)
`Report info's/warnings/errors` check boxes (Macintosh)

Specifies the types of return status codes from the CDF library that should be reported/displayed.

VMS, UNIX, & MS-DOS: `<types>` is a comma-separated list of zero or more of the following symbols: `errors`, `warnings`, or `informationals`. Note that these symbols can be truncated (e.g., `e`, `w`, and `i`).

`/CACHE=<size>` (VMS)
`-cache <size>` (UNIX & MS-DOS)
`Cache size` edit field (Macintosh)

Specifies the cache size to be used for the `.cdf` file of both the source and destination CDF (i.e., the only files if they are both single-file CDFs). If omitted, default cache sizes as chosen by the CDF library are used. The cache size is the number of 512-byte buffers to be used.

`/[NO]STATISTICS` (VMS)
`-[no]statistics` (UNIX & MS-DOS)
`Display statistics` check box (Macintosh)

Specifies whether or not caching statistics are displayed at the end of each conversion.

Example(s):

VMS:

```
$ CDFCONVERT CDF$SMPL:TEMPLATE0 TEMPLATE0X
$ CDFCONVERT/LOG/REPORT=(ERRORS) CDF$SMPL: USER_DISK:[USER.CDF]
$ CDFCONVERT CAC_SST_BLENDED CAC_SST_BLENDEDX/SINGLE/NETWORK
```

```
$ CDFCONVERT/SKELETON=CDF$SMPL:TEMPLATE3 CAC_SST_BLENDED* [USER.CDF]
```

UNIX:

```
% cdfconvert ../samples/template0 template0x
% cdfconvert -log -report "errors" ../samples /disk4/user/cdf
% cdfconvert cac_sst_blended cac_sst_1 -single -network
% cdfconvert -skeleton template3 '../cdf/cac_sst*' ~user/cdf
```

MS-DOS:

```
> cdfconvert ..\samples\tplate0 tplate0x
> cdfconvert -log -report "errors" ..\samples c:\dir4\user\cdf
> cdfconvert cac_sst cac_sst1 -single -network
> cdfconvert -skeleton tplate3 a:\cdf\cac_sst dir5\cdf
```

VMS, UNIX, & MS-DOS: Command line help is displayed when `CDFconvert` is executed without any arguments.

3.4.3 Output from the CDFconvert Program

As `CDFconvert` executes, the name of each CDF being converted is displayed. If message logging is enabled, the progress of each conversion is also displayed.

3.5 CDFstats

3.5.1 Introduction

The `CDFstats` program produces a statistical report on a CDF's variable data. Both rVariables and zVariables are analyzed. For each variable it determines the actual minimum and maximum values (in all of the variable records), the minimum and maximum values within the valid range (if the `VALIDMIN` and `VALIDMAX` vAttributes and corresponding entries are present in the CDF), and the monotonicity. An option exists to allow fill values (specified by the `FILLVAL` vAttribute) to be ignored when collecting statistics.

Monotonicity refers to whether or not variable data values increase or decrease from record to record or along a dimension. This property is checked only if the variable varies along just one "dimension" (considering records to be another "dimension"). For example, consider a CDF with the 2-dimensional rVariables shown in Table 3.1.

The `EPOCH`, `LATITUDE`, and `LONGITUDE` rVariables would be checked for monotonicity but the `ELEVATION` and `TEMPERATURE` rVariables would not be checked.

An additional option exists that allows the `VALIDMIN`, `VALIDMAX`, `SCALEMIN`, and `SCALEMAX` attribute entries to be updated based on the actual minimum and maximum values found for a variable. The `MONOTON` attribute entries may also be updated based on the monotonicity found for a variable.

rVariable	Record Variance	Dimension Variances
EPOCH	VARY	NOVARY,NOVARY
LATITUDE	NOVARY	VARY,NOVARY
LONGITUDE	NOVARY	NOVARY,VARY
ELEVATION	NOVARY	VARY,VARY
TEMPERATURE	VARY	VARY,VARY

Table 3.1: Example rVariables, CDFstats Monotonicity Checking

3.5.2 Executing the CDFstats Program

Usage:

VMS:

```
$ CDFSTATS [/[NO]RANGE] [/[NO]FILL] [/OUTPUT=<file-path>] [/[NO]FORMAT]
           [/[NO]PAGE] [/[NO]UPDATE_VALIDS] [/[NO]UPDATE_SCALES]
           [/[NO]UPDATE_MONOTONIC] [/ZMODE=<mode>] [/[NO]NEG2POSFP0]
           [/REPORT=(<types>)] [/CACHE=<size>] <cdf-path>
```

UNIX:

```
% cdfstats [-[no]range] [-[no]fill] [-output <file-name>] [-[no]format]
           [-[no]page] [-[no]update_valids] [-[no]update_scales]
           [-[no]update_monotonic] [-zmode <mode>] [-[no]neg2posfp0]
           [-report "<types>"] [-cache <size>] <cdf-path>
```

MS-DOS:

```
> cdfstats [-[no]range] [-[no]fill] [-output <file-name>] [-[no]format]
           [-[no]page] [-[no]update_valids] [-[no]update_scales]
           [-[no]update_monotonic] [-zmode <mode>] [-[no]neg2posfp0]
           [-report "<types>"] [-cache <size>] <cdf-path>
```

Macintosh:

Double-click on the CDFstats icon. When the desired parameters/qualifiers have been selected in the dialog box, click on the Enter button to analyze the CDF. Clicking on the Help button will display online help. Clicking on the Quit button terminates CDFstats.

Parameter(s):

<cdf-path> (VMS, UNIX, & MS-DOS)
CDF edit field (Macintosh)

The pathname of the CDF to analyze. (Do not specify an extension.)

Macintosh: At the end of this field a button labeled `Select` is present. When selected, a standard input file dialog is displayed from which the CDF may be selected.

Qualifier(s):

`/[NO]RANGE` (VMS)
`-[no]range` (UNIX & MS-DOS)
`Range checking` check box (Macintosh)

Specifies whether or not range checking will be performed. To perform range checking, the CDF must contain `VALIDMIN` and `VALIDMAX` attributes. A variable must also have an entry for each of these attributes in order for range checking to be performed on that variable.

`/[NO]FILL` (VMS)
`-[no]fill` (UNIX & MS-DOS)
`Use FILLVAL` check box (Macintosh)

Specifies whether or not fill values are ignored when collecting statistics (using the `FILLVAL` attribute if it exists).

`/OUTPUT=<file-path>` (VMS)
`-output <file-path>` (UNIX & MS-DOS)
`Output` edit field (Macintosh)

If this qualifier is specified, the statistical output is written to the named file. If the named file does not have an extension, `.sts` (UNIX & Macintosh) or `.STS` (VMS & MS-DOS) is appended automatically. If this qualifier is not specified, the output is displayed on the screen.

Macintosh: At the end of this field a button labeled `Select` is present. When selected, a standard output file dialog is displayed in which the output file may be specified.

`/[NO]FORMAT` (VMS)
`-[no]format` (UNIX & MS-DOS)
`Use FORMAT` check box (Macintosh)

Specifies whether or not the `FORMAT` attribute is used when displaying variable values (if the `FORMAT` attribute exists and an entry exists for the variable).

`/[NO]PAGE` (VMS)
`-[no]page` (UNIX & MS-DOS)
`Page output` check box (Macintosh)

Specifies whether or not the output is displayed a page (screen) at a time. If the output is directed to a file, paging is not allowed.

`/[NO]UPDATE_VALIDS` (VMS)
`-[no]update_valids` (UNIX & MS-DOS)
`Update VALIDMIN/MAX` check boxes (Macintosh)

Specifies whether or not the `VALIDMIN` and `VALIDMAX` attribute entry values are updated for each variable based on the actual minimum and maximum values found (with fill values being ignored if requested). If the `VALIDMIN` and `VALIDMAX` attributes do not exist, they are created.

`/[NO]UPDATE_SCALES` (VMS)

-[no]update_scales (UNIX & MS-DOS)
Update SCALEMIN/MAX check boxes (Macintosh)

Specifies whether or not the SCALEMIN and SCALEMAX attribute entry values are updated for each variable based on the actual minimum and maximum values found (with fill values being ignored if requested). If the SCALEMIN and SCALEMAX attributes do not exist, they are created.

/[NO]UPDATE_MONOTONIC (VMS)
-[no]update_monotonic (UNIX & MS-DOS)
Update MONOTON check box (Macintosh)

Specifies whether or not the MONOTONIC attribute entry values are updated for each variable based on the monotonicity found (with fill values being ignored if requested). If the MONOTONIC attribute does not exist, it is created.

/ZMODE=<mode> (VMS)
-zmode <mode> (UNIX & MS-DOS)
zMode radio buttons (Macintosh)

Specifies the zMode that should be used with the CDF. The zMode may be one of the following:

0 Indicates that zMode should be disabled.

1 Indicates that zMode/1 should be used. (The dimension variances of rVariables will be preserved.)

2 Indicates that zMode/2 should be used. (The dimensions of rVariables having a variance of NOVARY [false] are removed.)

/[NO]NEG2POSFP0 (VMS)
-[no]neg2posfp0 (UNIX & MS-DOS)
-0.0 to 0.0 check box (Macintosh)

Specifies whether or not -0.0 is converted to 0.0 by the CDF library when read from a CDF. -0.0 is an illegal floating point value on VAXes and DEC Alphas running OpenVMS.

/REPORT=(<types>) (VMS)
-report "<types>" (UNIX & MS-DOS)
Report info's/warnings/errors check boxes (Macintosh)

Specifies the types of return status codes from the CDF library that should be reported/displayed.

VMS, UNIX, & MS-DOS: <types> is a comma-separated list of zero or more of the following symbols: errors, warnings, or informationals. Note that these symbols can be truncated (e.g., e, w, and i).

/CACHE=<size> (VMS)
-cache <size> (UNIX & MS-DOS)
Cache size edit field (Macintosh)

Specifies the cache size to be used for the CDF's .cdf file (i.e., the only file if a single-file CDF). If omitted, a default cache size as chosen by the CDF library is used. The cache size is the number of 512-byte buffers to be used.

Example(s):

VMS:

```
$ CDFSTATS TEST1
$ CDFSTATS/REPORT=(ERRORS) GISS_SOIL
$ CDFSTATS/NOFILL/OUTPUT=TEMPLATE3/NORANGE CDF$SMPL:TEMPLATE3
```

UNIX:

```
% cdfstats giss_soil
% cdfstats -range -fill -report "errors" $CDF_SMPL/giss_soil
% cdfstats -norange -output template3 ../../samples/template3
```

MS-DOS:

```
> cdfstats gisssoil
> cdfstats -range -nofill -report "errors" a:\cdfs\gisssoil
> cdfstats -norange -output tplate3 ..\..\samples\tplate3
```

VMS, UNIX, & MS-DOS: Command line help is displayed when `CDFstats` is executed without any arguments.

3.5.3 Output from the CDFstats Program

The format of the output from `CDFstats` is as follows:

For each variable (rVariables and zVariables),

```
<number>.  <name>  <n-dims>:[<dim-sizes>]  <rec-vary>/<dim-varys>  (<data-type>/<n-elems>)

              min: <min-value>
     min in range: <min-value-in-range>
        valid min: <valid-min>, <low-values> low value(s)

              max: <max-value>
     max in range: <max-value-in-range>
        valid max: <valid-max>, <high-values> high value(s)

       fill value: <fill-value>, <fill-values> fill value(s)

        monotonic: <monotonicity>
```

If range checking and/or fill value filtering is disabled, the corresponding fields will not be displayed. The fields are defined as follows:

`<number>`	The variable number.
`<name>`	The variable name.
`<rec-vary>`	The record variance of the variable — either a `T` or `F`.
`<dim-varys>`	The dimension variances of the variable — for each dimension either a `T` or `F`. This field is not present if there are zero (0) dimensions.
`<data-type>`	The data type of the variable (e.g., `CDF_REAL4`).
`<n-elems>`	The number of elements of the variable's data type.
`<n-dims>`	The number of dimensions of a zVariable. This field is not present for an rVariable.
`<dim-sizes>`	The dimension sizes of a zVariable. This field is not present for an rVariable or if the zVariable has zero (0) dimensions.
`<min-value>`	The minimum value found (regardless of any range checking performed).
`<min-value-in-range>`	The minimum value found within the valid range.
`<valid-min>`	The minimum valid value (`VALIDMIN` attribute entry value).
`<low-values>`	The number of values found that are less than the valid minimum.
`<max-value>`	The maximum value found (regardless of any range checking performed).
`<max-value-in-range>`	The maximum value found with the valid range.
`<valid-max>`	The maximum valid value (`VALIDMAX` attribute entry value).
`<high-values>`	The number of values found that are greater than the valid maximum.
`<fill-value>`	The fill value (`FILLVAL` attribute entry value).
`<fill-values>`	The number of fill values found.
`<monotonicity>`	The monotonicity of the variable.

The `<monotonicity>` field may take on one of the following values.

`Steady (one value)`	The variable has only one value in the CDF.
`Steady (all values the same)`	All values of the variable are the same.
`Increase`	Values strictly increase (with increasing record number/dimension index).
`Decrease`	Values strictly decrease (with increasing record number/dimension index).
`noDecrease (some values the same)`	Consecutive values either increase or are the same (with increasing record number/dimension index).

`noIncrease (some values the same)`	Consecutive values either decrease or are the same (with increasing record number/dimension index).
`False`	Consecutive values both increase and decrease.
`n/a`	The variable was not checked for monotonicity because it varies along more than one "dimension" (if records are considered another "dimension").

3.6 CDFinquire

3.6.1 Introduction

The `CDFinquire` program displays the version of the CDF distribution being used and the default toolkit qualifiers.[5]

3.6.2 Executing the CDFinquire Program

Usage:

VMS:

```
$ CDFINQUIRE [/ID] [/[NO]PAGE]
```

UNIX:

```
% cdfinquire [-id] [-[no]page]
```

MS-DOS:[6]

```
> cdfinquire [-id] [-[no]page]
```

Macintosh:

Double-click on the `CDFinquire` icon. When the desired parameters/qualifiers have been selected in the dialog box, click on the `Enter` button to inquire the CDF distribution. Clicking on the `Help` button will display online help. Clicking on the `Quit` button terminates `CDFinquire`.

[5] In CDF V2.5, the previous functionality of `CDFinquire` has been moved to the `SkeletonTable` toolkit program.

[6] On MS-DOS systems the executable is named `CDFINQ.EXE`.

Parameter(s):

`(None)`

Qualifier(s):

`/ID` (VMS)
`-id` (UNIX & MS-DOS)
`Id` check box (Macintosh)

Causes the version of your CDF distribution and the default toolkit qualifiers to be displayed. This qualifier is required.

`/[NO]PAGE` (VMS)
`-[no]page` (UNIX & MS-DOS)
`Page output` check box (Macintosh)

Specifies whether or not the output is displayed a page (screen) at a time.

Example(s):

VMS:

```
$ CDFINQUIRE/ID/PAGE
```

UNIX:

```
% cdfinquire -id -page
```

MS-DOS:

```
> cdfinquire -id -page
```

VMS, UNIX, & MS-DOS: Command line help is displayed when `CDFinquire` is executed without any arguments.

3.6.3 Output from the CDFinquire Program

The version of your CDF distribution is displayed first followed by the default toolkit qualifiers in the style of the operating system being used.

3.7 SkeletonTable

3.7.1 Introduction

The `SkeletonTable` program is used to create an ASCII text file called a skeleton table containing information about a given CDF. (`SkeletonTable` can also be instructed to output the skeleton table to the terminal screen.) It reads a CDF and writes into the skeleton table the following information.

1. Format (single or multi file), data encoding, variable majority.
2. Number of dimensions and dimension sizes for the rVariables.
3. gAttribute definitions (and gEntry values).
4. rVariable and zVariable definitions and vAttribute definitions (with rEntry/zEntry values).
5. Data values for all or a subset of the CDF's variables. Traditionally, only NRV variable values are written to a skeleton table. RV variable values may now also be written. NRV and RV variables are described in Section 2.3.10.

The above information is written in a format that can be "understood" by the `SkeletonCDF` program (see Section 3.8). `SkeletonCDF` reads a skeleton table and creates a new CDF (called a skeleton CDF).

3.7.2 Executing the SkeletonTable Program

Usage:

VMS:

```
$ SKELETONTABLE [/SKELETON=<skeleton-path>] [/[NO]LOG] [/ZMODE <mode>]
                [/NONRV | /NRVTABLE | /VALUES=<values>] [/[NO]SCREEN]
                [-[NO]NEG2POSFP0] [/[NO]FORMAT] [/REPORT=(<types>)]
                [/CACHE=<size>] [/[NO]PAGE] <cdf-path>
```

UNIX:

```
% skeletontable [-skeleton <skeleton-path>] [-[no]log] [-zmode <mode>]
                [-nonrv | -nrvtable | -values <values>] [-[no]screen]
                [-[no]neg2posfp0] [-[no]format] [-report "<types>"]
                [-cache <size>] [-[no]page] <cdf-path>
```

MS-DOS:[7]

[7] On MS-DOS systems the executable is named `CDF2SKT.EXE`.

```
> skeletontable [-skeleton <skeleton-path>] [-[no]log] [-zmode <mode>]
                [-nonrv | -nrvtable | -values <values>] [-[no]screen]
                [-[no]neg2posfp0] [-[no]format] [-report "<types>"]
                [-cache <size>] [-[no]page] <cdf-path>
```

Macintosh:

Double-click on the `SkeletonTable` icon. When the desired parameters/qualifiers have been selected in the dialog box, click on the `Enter` button to create the skeleton table. Clicking on the `Help` button will display online help. Clicking on the `Quit` button terminates `SkeletonTable`.

Parameter(s):

`<cdf-path>` (VMS, UNIX, & MS-DOS)
`CDF` edit field (Macintosh)

The pathname of the CDF from which the skeleton table will be created. (Do not enter an extension.)

Macintosh: At the end of this field a button labeled `Select` is present. When selected, a standard input file dialog is displayed from which the CDF may be selected.

Qualifier(s):

`/SKELETON=<skeleton-path>` (VMS)
`-skeleton <skeleton-path>` (UNIX & MS-DOS)
`Skeleton` edit field (Macintosh)

The pathname of the skeleton table to be created. (Do not enter an extension because `.skt` [UNIX & Macintosh] or `.SKT` [VMS & MS-DOS] is appended automatically.) If this qualifier is not specified, the skeleton table will be named `<cdf-name>.skt` in the default/current directory (where `<cdf-name>` is the name portion of the CDF from which the skeleton table was created).

Macintosh: At the end of this field a button labeled `Select` is present. When selected, a standard output file dialog is displayed in which the skeleton table may be specified.

`/VALUES=<values> | /NRVTABLE | /NONRV` (VMS)
`-values <values> | -nrvtable | -nonrv` (UNIX & MS-DOS)
`No values/.../Selected values` radio buttons (Macintosh)

Only one of these qualifiers may be specified. The meaning of each is as follows:

`/VALUES=<values>` (VMS)
`-values <values>` (UNIX & MS-DOS)
`No values/.../Selected values` radio buttons (Macintosh)

VMS, UNIX, & MS-DOS: The `<values>` option specifies which variable values should be put in the skeleton table. Select one of the options from the list which follows. **Macintosh:** Selecting one of the radio buttons described in the following list specifies which variable values should be put in the skeleton table.

`none` (VMS, UNIX, & MS-DOS)
`No values` radio button (Macintosh)

No variable values should be put in the skeleton table.

`nrv` (VMS, UNIX, & MS-DOS)
`NRV values` radio button (Macintosh)

Only NRV variable values should be put in the skeleton table.

`rv` (VMS, UNIX, & MS-DOS)
`RV values` radio button (Macintosh)

Only RV variable values should be put in the skeleton table.

`all` (VMS, UNIX, & MS-DOS)
`All values` radio button (Macintosh)

All variable values should be put in the skeleton table.

`<named>` (VMS, UNIX, & MS-DOS)
`Selected values` radio button (Macintosh)

Values of the named variables should be put in the skeleton table.

VMS, UNIX, & MS-DOS: `<values>` is a comma-separated list of delimited variable names with the entire list enclosed in double quote marks. **NOTE:** Do not use double quote marks to delimit a variable name.

Macintosh: The named variables are specified in the `Variables` edit field as a comma-separated list of delimited variable names.

`/NONRV` (VMS)
`-nonrv` (UNIX & MS-DOS)
Not supported (Macintosh)

Ignore NRV data. (No values are not placed in the skeleton table.)

`/NRVTABLE` (VMS)
`-nrvtable` (UNIX & MS-DOS)
Not supported (Macintosh)

Put NRV variable data values in the skeleton table.

VMS, UNIX, & MS-DOS: Note that only the "values" qualifier is actually needed. The others are supported for compatibility with previous CDF distributions.

`/[NO]LOG` (VMS)
`-[no]log` (UNIX & MS-DOS)
`Log progress` check box (Macintosh)

Specifies whether or not messages are displayed as the program executes.

`/ZMODE=<mode>` (VMS)
`-zmode <mode>` (UNIX & MS-DOS)
`zMode` radio buttons (Macintosh)

Specifies the zMode that should be used with the CDF. The zMode may be one of the following:

`0` Indicates that zMode should be disabled.

`1` Indicates that zMode/1 should be used. (The dimension variances of rVariables will be preserved.)

`2` Indicates that zMode/2 should be used. (The dimensions of rVariables having a variance of `NOVARY` [false] are removed.)

`/[NO]FORMAT` (VMS)
`-[no]format` (UNIX & MS-DOS)
`Use FORMAT` check box (Macintosh)

Specifies whether or not the `FORMAT` attribute is used when writing variable values (if the `FORMAT` attribute exists and an entry exists for the variable).

`/[NO]NEG2POSFP0` (VMS)
`-[no]neg2posfp0` (UNIX & MS-DOS)
`-0.0 to 0.0` check box (Macintosh)

Specifies whether or not `-0.0` is converted to `0.0` by the CDF library when read from a CDF. `-0.0` is an illegal floating point value on VAXes and DEC Alphas running OpenVMS.

`/REPORT=(<types>)` (VMS)
`-report "<types>"` (UNIX & MS-DOS)
`Report info's/warnings/errors` check boxes (Macintosh)

Specifies the types of return status codes from the CDF library that should be reported/displayed.

VMS, UNIX, & MS-DOS: `<types>` is a comma-separated list of zero or more of the following symbols: `errors`, `warnings`, or `informationals`. Note that these symbols can be truncated (e.g., `e`, `w`, and `i`).

`/CACHE=<size>` (VMS)
`-cache <size>` (UNIX & MS-DOS)
`Cache size` edit field (Macintosh)

Specifies the cache size to be used for the CDF's `.cdf` file (i.e., the only file if a single-file CDF). If omitted, a default cache size as chosen by the CDF library is used. The cache size is the number of 512-byte buffers to be used.

`/[NO]SCREEN` (VMS)
`-[no]screen` (UNIX & MS-DOS)
`Output to screen` check box (Macintosh)

Specifies whether or not the skeleton table is to be displayed on the terminal screen (written to the "standard output"). If not, the skeleton table is written to a text file.

`/[NO]PAGE` (VMS)
`-[no]page` (UNIX & MS-DOS)
`Page output` check box (Macintosh)

If the skeleton table is being displayed on the terminal screen, specifies whether or not the output is displayed one page (screen) at a time.

Example(s):

VMS:

```
$ SKELETONTABLE/NOLOG/REPORT=(ERRORS) FGGE3B
$ SKELETONTABLE/SKELETON=FGGE3B/NONRV FGGE3B
$ SKELETONTABLE/SCREEN/VALUES="'Var1','Var2'"
```

UNIX:

```
% skeletontable -nolog -report "errors" fgge3b
% skeletontable -skeleton fgge3b -nonrv ../cdfs/fgge3b
% skeletontable -values "'Var1','Var2'"
```

MS-DOS:

```
> skeletontable -nolog -report "errors" fgge3b
> skeletontable -skeleton fgge3b -nonrv c:\temp\fgge3b
> skeletontable -values "'Var1','Var2'"
```

VMS, UNIX, & MS-DOS: Command line help is displayed when `SkeletonTable` is executed without any arguments.

3.7.3 Output from the SkeletonTable Program

The format of the skeleton table is described in Appendix A.

3.8 SkeletonCDF

3.8.1 Introduction

The `SkeletonCDF` [8] program is used to make a fully structured CDF, called a skeleton CDF, by reading a text file called a skeleton table. The skeleton table contains the information necessary to create a CDF that is complete in all respects except for record-variant (RV) variable values. (RV variables vary from record to record.) RV values are then written to the CDF by the execution of an application program. The `SkeletonCDF` program allows a CDF to be created with the following.

1. The necessary header information — the number of dimensions and dimension sizes for the rVariables, format, data encoding, and variable majority.

[8] This program was originally named `CDFskeleton`. It has been renamed to ease the confusion caused some users. Now, `SkeletonCDF` is used to create skeleton CDFs and `SkeletonTable` is used to create skeleton tables.

2. The gAttribute definitions and any number of gEntries for each.

3. The rVariable and zVariable definitions.

4. The vAttribute definitions and the entries corresponding to each variable.

5. The data values for those variables that are non-record-variant (NRV). NRV variables do not vary from record to record.[9]

The created CDF is referred to as a skeleton CDF.

3.8.2 Executing the SkeletonCDF Program

Usage:

VMS:

```
$ SKELETONCDF [/CDF=<cdf-path>] [/[NO]LOG] [/[NO]DELETE] [/[NO]FILLVAL]
              [/REPORT=(<types>)] [/[NO]NEG2POSFP0] <skeleton-path>
```

UNIX:

```
% skeletoncdf [-cdf <cdf-path>] [-[no]log] [-[no]delete] [-[no]fillval]
              [-report "<types>"] [-[no]neg2posfp0] <skeleton-path>
```

MS-DOS:[10]

```
> skeletoncdf [-cdf <cdf-path>] [-[no]log] [-[no]delete] [-[no]fillval]
              [-report "<types>"] [-[no]neg2posfp0] <skeleton-path>
```

Macintosh:

Double-click on the `SkeletonCDF` icon. When the desired parameters/qualifiers have been selected in the dialog box, click on the `Enter` button to create the skeleton CDF. Clicking on the `Help` button will display online help. Clicking on the `Quit` button terminates `SkeletonCDF`.

Parameter(s):

<u>`<skeleton-path>` (VMS, UNIX, & MS-DOS)</u>

[9] RV variable data may also be specified in a skeleton table but this is generally not done given the amount of RV data typically contained in a CDF.

[10] On MS-DOS systems the executable is named `SKT2CDF.EXE`.

`Skeleton` edit field (Macintosh)

The pathname of the skeleton table from which a skeleton CDF will be created. (Do not specify an extension.)

Macintosh: At the end of this field a button labeled `Select` is present. When selected, a standard input file dialog is displayed from which a skeleton table may be selected.

Qualifier(s):

`/CDF=<cdf-path>` (VMS)
`-cdf <cdf-path>` (UNIX & MS-DOS)
`CDF` edit field (Macintosh)

The pathname of the CDF that will be created (overriding the pathname in the skeleton table). If this qualifier is not specified, the CDF pathname in the skeleton table is used. Do not specify an extension in the pathname.

Macintosh: At the end of this field a button labeled `Select` is present. When selected, a standard output file dialog is displayed in which a CDF name may be specified.

`/[NO]LOG` (VMS)
`-[no]log` (UNIX & MS-DOS)
`Log progress` check box (Macintosh)

Specifies whether or not messages are displayed as the program executes.

`/[NO]NEG2POSFP0` (VMS)
`-[no]neg2posfp0` (UNIX & MS-DOS)
`-0.0 to 0.0` check box (Macintosh)

Specifies whether or not `-0.0` is converted to `0.0` by the CDF library when written to a CDF. `-0.0` is an illegal floating point value on VAXes and DEC Alphas running OpenVMS.

`/[NO]DELETE` (VMS)
`-[no]delete` (UNIX & MS-DOS)
`Delete existing` check box (Macintosh)

Specifies whether or not the CDF will be deleted first if it already exists (essentially overwriting it).

`/[NO]FILLVAL` (VMS)
`-[no]fillval` (UNIX & MS-DOS)
`Use FILLVAL` check box (Macintosh)

Specifies whether or not entries of the `FILLVAL` vAttribute are used to set the pad values for the corresponding variables. If this qualifier is specified, the `FILLVAL` vAttribute must exist and only those variables with an entry for the `FILLVAL` vAttribute will be affected.

`/REPORT=(<types>)` (VMS)
`-report "<types>"` (UNIX & MS-DOS)
`Report info's/warnings/errors` check boxes (Macintosh)

Specifies the types of return status codes from the CDF library that should be reported/displayed.

VMS, UNIX, & MS-DOS: `<types>` is a comma-separated list of zero or more of the following symbols: `errors`, `warnings`, or `informationals`. Note that these symbols can be truncated (e.g., `e`, `w`, and `i`).

Example(s):

VMS:

```
$ SKELETONCDF FGGE3B
$ SKELETONCDF/NOLOG/CDF=[-.TEMP]FGGE3B_X/REPORT=(ERRORS) FGGE3B
```

UNIX:

```
% skeletoncdf fgge3b
% skeletoncdf -nolog -cdf ../fgge3b_x -report "errors" fgge3b
```

MS-DOS:

```
> skeletoncdf fgge3b
> skeletoncdf -nolog -cdf ..\fgge3b_x -report "errors" a:\fgge3b
```

VMS, UNIX, & MS-DOS: Command line help is displayed when `SkeletonCDF` is executed without any arguments.

3.8.3 Creating the Skeleton Table

A skeleton table is a text file having `.skt` (UNIX & Macintosh) or `.SKT` (VMS & MS-DOS) as a file extension. The normal method of creating and using a skeleton table would be to use `SkeletonTable` on an existing CDF that is similar to the CDF you want to create. Then edit the created skeleton table to meet your needs, and use `SkeletonCDF` to create the new CDF. The skeleton table could also be created from scratch with any text editor.

The format of the skeleton table is described in Appendix A.

3.9 CDFbrowse

`CDFbrowse` has been replaced by `CDFedit` (see Section 3.12).[11]

[11] The alias/symbol `CDFbrowse` still exists in the "definitions" file on UNIX/VMS systems but now executes `CDFedit` in a browse-only mode.

3.10 CDFlist

3.10.1 Introduction

The `CDFlist` program allows you to selectively list all (or a subset of) the variable data values from a CDF. The values can be displayed at the screen, output to a text file, or used to create a new CDF. (The new CDF will most likely be a subset of the original CDF.)

3.10.2 Executing the CDFlist Program

Usage:

VMS:

```
$ CDFLIST [/INITIAL=(<defaults>)] [/SPECIFICATION=<default-cdf-spec>]
          [/CACHE=<size>] [<initial-cdf-spec>]
```

UNIX:

```
% cdflist [-initial "<defaults>"] [-specification <default-cdf-spec>]
          [-cache <size>] [<initial-cdf-spec>]
```

MS-DOS:

```
> cdflist [-initial "<defaults>"] [-specification <default-cdf-spec>]
          [-cache <size>] [<initial-cdf-spec>]
```

Macintosh:

Double-click on the `CDFlist` icon. When the desired parameters/qualifiers have been selected in the dialog box, click on the `Enter` button to begin listing the CDF(s). Clicking on the `Help` button will display online help. Clicking on the `Quit` button terminates `CDFlist`.

Parameter(s):

`<initial-cdf-spec>` (VMS, UNIX, & MS-DOS)
`CDF` edit field (Macintosh)

The initial specification of the CDF(s) to list. (Do not specify an extension.) This may be either a single CDF pathname or a directory/wildcard path. Wildcards are allowed in the CDF name but not in the directory path. If this parameter is specified, then the "specification" qualifier may not be specified. If neither this parameter nor the "specification" qualifier are specified, then the initial CDF(s) specification will be the default/current directory.