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).

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.[26] 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

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

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 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 2.5. The entries for an attribute may have any combination of data specifications.

For character 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 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 Data Types

CDF supports a variety of data types consistent with the types available with C and Fortran compilers on most computers. All data types are based on an 8-bit byte. The size of an element of a data type is the same regardless of the computer/operating system being used. The <GET_,DATATYPE_SIZE_> operation of the Internal Interface may be used to inquire the size in bytes of a particular data type.

2.5.1 Integer Data Types

CDF_BYTE	1-byte, signed integer.
CDF_INT1	1-byte, signed integer.
CDF_UINT1	1-byte, unsigned integer.
CDF_INT2	2-byte, signed integer.
CDF_UINT2	2-byte, unsigned integer.
CDF_INT4	4-byte, signed integer.
CDF_UINT4	4-byte, unsigned integer.

NOTE: When using C on a DEC Alpha running OSF/1, keep in mind that a `long` is 8 bytes and that an `int` is 4 bytes. Use an `int` with the data types CDF_INT4 and CDF_UINT4 rather than a `long`.

NOTE: When using C on a PC under MS-DOS, keep in mind that an `int` is most likely 2 bytes and that a `long` is 4 bytes. Use a `long` with the data types `CDF_INT4` and `CDF_UINT4` rather than an `int`.

2.5.2 Floating Point Data Types

`CDF_REAL4 & CDF_FLOAT`	4-byte, single-precision floating-point.
`CDF_REAL8 & CDF_DOUBLE`	8-byte, double-precision floating-point.

A special case exists with respect to the value `-0.0` (negative floating-point zero). This value is legal on those computers that use the IEEE 754 floating-point representation (e.g., most UNIX-based computers and the PC) but is illegal on VAXes and DEC Alphas running OpenVMS. Attempting to use `-0.0` will result in a reserved operand fault on a VAX and a high performance arithmetic fault on a DEC Alpha running OpenVMS. A warning is returned whenever `-0.0` is read by an application on a VAX or DEC Alpha running OpenVMS. The CDF library can be put into a mode where `-0.0` will be converted to `0.0` when detected (see Section 2.1.2). If `-0.0` is not being converted to `0.0`, the CDF toolkit programs are designed to display `-0.0` in all cases. This includes those computers that normally suppress the negative sign.

2.5.3 Character Data Types

`CDF_CHAR`	1-byte, character.
`CDF_UCHAR`	1-byte, unsigned character.

Character data types are unique for variables in that they are the only data types for which more than one element per value is allowed. Each variable value consists of a character string with the number of elements being the number of characters. More than one element is allowed for any of the data types when dealing with attribute entries.

2.5.4 EPOCH Data Type

`CDF_EPOCH`	8-byte, double precision floating point.

The `CDF_EPOCH` data type is used to store time values referenced from a particular epoch. For NSSDC applications that epoch is `01-Jan-0000 00:00:00.000`.[27] `CDF_EPOCH` values are the number of milliseconds since the epoch. The standard format used to display a `CDF_EPOCH` value is...

```
dd-mmm-yyyy hh:mm:ss.ccc
```

where `dd` is the day of the month (01-31), `mmm` is the month (`Jan`, `Feb`, `Mar`, `Apr`, `May`, `Jun`, `Jul`, `Aug`, `Sep`, `Oct`, `Nov`, or `Dec`), `yyyy` is the year (0000-9999) `hh` is the hour (00-23), `mm` is the minute (00-59), `ss` is the second (00-59), and `ccc` is the millisecond (000-999).

[27] I know what you're thinking. The year 0 AD never existed. If it makes you feel better, think of the epoch year as 1 BC (or simply year 0) rather than 0 AD. Also, year 0 is considered to be a leap year.

Functions exist that parse, encode, compute, and decompose CDF_EPOCH values. These functions are described in the CDF C Reference Manual for C applications and in the CDF Fortran Reference Manual for Fortran applications.

2.5.5 Equivalent Data Types

Certain data types are considered equivalent with respect to their representation in memory and in a CDF. Table 2.9 shows the groups of equivalent data types.

CDF_CHAR	CDF_INT2	CDF_INT4	CDF_REAL4	CDF_REAL8
CDF_UCHAR	CDF_UINT2	CDF_UINT4	CDF_FLOAT	CDF_DOUBLE
CDF_INT1				CDF_EPOCH
CDF_UINT1				
CDF_BYTE				

Table 2.9: Equivalent Data Types

Note that while the signed and unsigned forms of a data type are considered equivalent by the CDF library, they must be correctly interpreted by an application to produce the desired results.

2.6 Compression Algorithms

Several compression algorithms are supported by the CDF library. Selecting the proper algorithm to use will depend on the characteristics of the data being compressed. Experimentation with the available algorithms on the CDF or variable being compressed will also be necessary. The following sections describe each compression algorithm, any associated parameters, and the types of data for which they are appropriate.

2.6.1 Run-Length Encoding

The run-length encoding compression algorithm, RLE_COMPRESSION, takes advantage of repeating bytes in the data. Currently, only the run-length encoding of zeros (0's) is supported. RLE_COMPRESSION has one parameter which must be set to RLE_OF_ZEROs. The notation RLE.0 is used for this type of RLE compression.

2.6.2 Huffman

The Huffman compression algorithm, HUFF_COMPRESSION, takes advantage of the frequency at which certain byte values occur in the data. A sequence of bytes that contain a high percentage of a limited number of byte values will compress better than if each byte value occurs with equal probability. HUFF_COMPRESSION has one parameter which must be set to OPTIMAL_ENCODING_TREES.[28] The notation HUFF.0 is used for this type of HUFF compression.

[28] OPTIMAL_ENCODING_TREES causes each buffer of data to be scanned for the best possible compression. An alternative method would be to scan the first buffer being compressed and then use the same byte value frequencies for subsequent buffers.

2.6.3 Adaptive Huffman

The adaptive Huffman compression algorithm, AHUFF_COMPRESSION, also takes advantage of the frequency at which certain byte values occur in the data. AHUFF_COMPRESSION is very similar to HUFF_COMPRESSION and generally provides slightly better compression. AHUFF_COMPRESSION has one parameter which must be set to OPTIMAL_ENCODING_TREES.[28] The notation AHUFF.0 is used for this type of AHUFF compression.

2.6.4 GZIP

The Gnu ZIP compression algorithm, GZIP_COMPRESSION, uses the Lempel-Ziv coding (LZ77) taking advantage of common substrings within the data. Significant compression occurs over a wide variety of data sets. GZIP_COMPRESSION has one parameter which may be set to a level value in the range from 1 (one) to 9 (nine). 1 provides the least amount of compression and executes the fastest. 9 provides the most compression but executes the slowest. Levels between 1 and 9 allow for a trade-off between compression and execution speed. The notation GZIP.<level> is used for GZIP compression where <level> is a value from 1 to 9. For example, GZIP.7 specifies a level of 7.

Note: GZIP compression is disabled for PCs running 16-bit DOS/Windows 3.x due to their memory constraint.

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 or CDFexport.

12. Some of the toolkit programs have a "paging" qualifier. Paging is not allowed if the output of the program has been directed to a file.

13. Most toolkit programs have an "about" qualifier that can be used to determine the CDF distribution from which the program came. On the Macintosh, an "about" selection is available on the "apple" pull-down menu.

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 systems you should have `source`'d (or equivalent) the script file named `definitions.<shell-type>` where `<shell-type>` is the type of shell you are using: `C` for the C-shell (`csh`) and `tcsh`, `K` for the Korn (`ksh`), BASH, and POSIX shells, and `B` for the Bourne shell (`sh`). This will define the necessary environment variables and aliases. Your system administrator knows the location of `definitions.<shell-type>`.

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 and CDFexport), 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

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

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 particular 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.

3.1.3 Special Attributes

There is a set of vAttributes that have special meaning to some of the CDF toolkit programs.[2] Your CDFs do not have to use these special attributes. The CDF toolkit programs will function properly whether or not these special attributes are present in a CDF. How the entries of each vAttribute are used for the corresponding variables is as follows:

`FORMAT`	A Fortran or C format specification that is used when displaying a variable value.
`VALIDMIN`	The minimum valid value for a variable.
`VALIDMAX`	The maximum valid value for a variable.
`FILLVAL`	The value used for missing or invalid variable values.[3]
`MONOTON`	The monotonicity of a variable: `INCREASE` (strictly increasing values), `DECREASE` (strictly decreasing values), or `FALSE` (not monotonic). Monotonicity only applies to NRV variables that vary along one dimension and RV variables that vary along no dimensions.
`SCALEMIN`	The minimum value for scaling a variable when graphically displaying its values.
`SCALEMAX`	The maximum value for scaling a variable when graphically displaying its values.

In the description of each CDF toolkit program, the special attributes that may affect that program's operation are defined. Note that most of the CDF toolkit programs can be instructed to ignore these special attributes.

3.1.4 Special Qualifier

There is a special qualifier applied to all toolkit programs. This qualifier, as `"-about"` on all platforms except Macintosh, will show version, release and increment information of the distribution that the toolkit

[2] These special attributes originated as part of the NSSDC standard for CDFs. The NSSDC standard is no longer used.

[3] Note that the `FILLVAL` attribute is not the same as the pad value for a variable although their values will often be the same. The pad value is used by the CDF library. The `FILLVAL` attribute is optionally used by a CDF toolkit program or by your applications.

program is based on. This special qualifier, if present, supersedes all other qualifiers and parameters.

3.2 CDFedit

3.2.1 Introduction

The `CDFedit` program allows the display and/or modification of practically all of the contents of a CDF by way of a full-screen interface. It is also possible to run CDFedit in a browse-only mode in order to prevent accidental modifications.[4]

3.2.2 Special Attribute Usage

The special attribute `FORMAT` is used by CDFedit (depending on the setting of the "format" qualifier) when displaying variable values.

3.2.3 Executing the CDFedit Program

Usage:

VMS:

```
$ CDFEDIT [/[NO]BROWSE] [/ZMODE=<mode>] [/[NO]FORMAT] [/[NO]PROMPT]
          [/[NO]NEG2POSFP0] [/REPORT=(<types>)] [/CACHE=(<sizes>)]
          [/[NO]STATISTICS] [/[NO]GWITHENTRIES] [/[NO]VWITHENTRIES]
          <cdf-spec>
```

UNIX:

```
% cdfedit [-[no]browse] [-zmode <mode>] [-[no]format] [-[no]prompt]
          [-[no]neg2posfp0] [-report "<types>"] [-cache "<sizes>"]
          [-[no]statistics] [-[no]gwithentries] [-[no]vwithentries]
          <cdf-spec>
```

MS-DOS:

```
> cdfedit [-[no]browse] [-zmode <mode>] [-[no]format] [-[no]prompt]
          [-[no]neg2posfp0] [-report "<types>"] [-cache "<sizes>"]
          [-[no]statistics] [-[no]gwithentries] [-[no]vwithentries]
          <cdf-spec>
```

[4] Running CDFedit in a browse-only mode provides the same functionality as CDFbrowse once did.

Macintosh:

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

Parameter(s):

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

The specification of the CDF(s) to edit. (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 the "prompt" qualifier is used, this will appear as the initial specification at the prompt. If this parameter is omitted, the "prompt" qualifier must be specified (and the initial specification at the prompt will be the default/current directory).

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. Also present is a button labeled `New`. When selected, a standard output file dialog is displayed in which a single CDF may be specified.

Qualifier(s):

`/[NO]BROWSE` (VMS)
`-[no]browse` (UNIX & MS-DOS)
`Browse only` check box (Macintosh)

Specifies whether or not a browsing mode is desired. In browsing mode the creation, modification, or deletion of a CDF is not allowed.

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

Specifies which zMode should be used. 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 displaying variable values (if the `FORMAT` attribute exists and an entry exists for the variable).

`/[NO]PROMPT` (VMS)

`-[no]prompt` (UNIX & MS-DOS)
`Prompt for CDF` check box (Macintosh)

Specifies whether or not a prompt is issued for the CDF(s) specification. When enabled the prompt will be issued both at program startup and after editing the current CDF(s) specification (at which point a new CDF[s] specification may be specified).

VMS, UNIX, & MS-DOS: If a CDF(s) specification was entered on the command line, that CDF(s) specification will appear at the prompt. (Otherwise, the current/default directory will appear at the prompt.)

Macintosh: If a CDF(s) specification was entered in the `CDF` edit field, that CDF(s) specification will appear at the prompt. (Otherwise, the current directory will appear at the prompt.)

`/[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 encountered in 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. The `<types>` option 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=(<sizes>)` (VMS)
`-cache "<sizes>"` (UNIX & MS-DOS)
`Cache sizes` edit field (Macintosh)

Specifies the cache sizes to be used by the CDF library for the dotCDF file and the various scratch files. The `<sizes>` option is a comma-separated list of `<size><type>` pairs where `<size>` is a cache size and `<type>` is the type of file as follows: `d` for the dotCDF file, `s` for the staging scratch file, and `c` for the compression scratch file. For example, `200d,100s` specifies a cache size of `200` for the dotCDF file and a cache size of `100` for the staging scratch file. The dotCDF file cache size can also be specified without the `d` file type for compatibility with older CDF releases (e.g., `200,100s`). Note that not all of the file types must be specified. Those not specified will receive a default cache size chosen by the CDF library. A cache size is the number of 512-byte buffers to be used. Section 2.1.5 explains the caching scheme used by the CDF library.

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

Specifies whether or not caching statistics are displayed when a CDF is closed.

`/[NO]GWITHENTRIES` (VMS)
`-[no]gwithentries` (UNIX & MS-DOS)
`gEntries with gAttributes` check box (Macintosh)

Specifies whether or not gEntries are displayed with the gAttributes or on separate menus (with one menu per gAttribute).

`/[NO]VWITHENTRIES` (VMS)

`-[no]vwithentries` (UNIX & MS-DOS)
`rEntries/zEntries with vAttributes` check box (Macintosh)

Specifies whether or not rEntries/zEntries are displayed with the vAttributes or on separate menus (with one menu per vAttribute).

Example(s):

VMS:

```
$ CDFEDIT [.SAMPLES]
$ CDFEDIT/ZMODE=2/NOFORMAT/CACHE=(10D,100S,200C) GISS_WETLX
$ CDFEDIT/BROWSE/PROMPT/REPORT=(ERRORS)
```

UNIX:

```
% cdfedit samples
% cdfedit -zmode 2 -noformat -cache "10d,100s,200c" giss_wetl
% cdfedit -browse -prompt -report "errors"
```

MS-DOS:

```
> cdfedit samples
> cdfedit -zmode 2 -noformat -cache "10d,100s,200c" giss_wetl
> cdfedit -browse -prompt -report "errors"
```

3.2.4 Interaction with CDFedit

Interaction with CDFedit is through a series of menus and windows. Extensive online help is provided and will not be repeated here.[5] The online help does refer to the sections of a window by name. Figure 3.1 illustrates the various sections of the possible types of windows.

ItemWindows are used when a choice is to be made from a list of one or more items (e.g., functions to perform, CDFs to edit, variable names, etc.). In some cases the entire list of items may not fit on the screen at once. When this occurs, the ItemSection may be scrolled to display hidden items. Some ItemWindows have a percentage indicator at the bottom right portion of the ItemSection. The percentage indicator shows which part of the ItemSection is being displayed.

PromptWindows are used when a textual response is required (e.g., a CDF specification, a new attribute name, a variable value, etc.). If the text is too long to fit into the PromptField, the "more" indicators ("<" and ">") at the left and right ends of the PromptField will display where hidden characters exist.

EditWindows are used to display/edit a text file or group of lines. EditWindows are currently used to display online help and to edit gAttribute character string entries as if they were a text file.

[5] It is our intention that the use of CDFedit be as intuitive as possible. You may not even need the online help. We're sure you'll let us know.

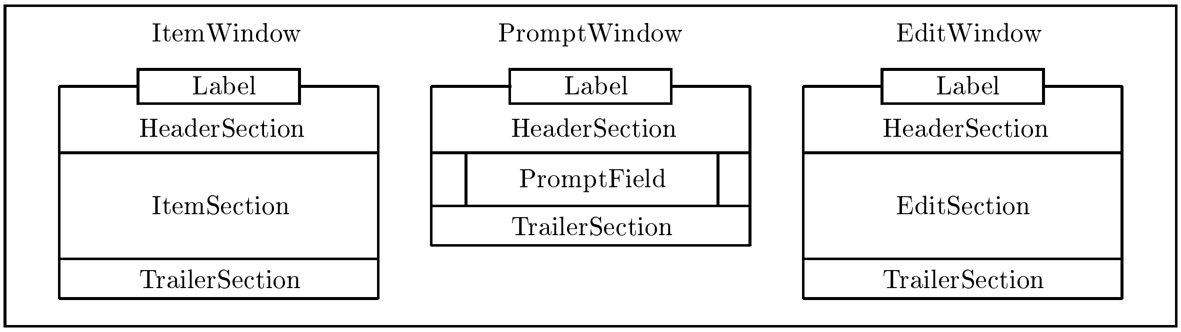

Figure 3.1: Window Sections, CDFedit

3.3 CDFexport

3.3.1 Introduction

CDFexport allows the contents of a CDF to be exported to the terminal screen, a text file, or another CDF. The variables to be exported can be selected along with a filter range for each variable which allows a subset of the CDF to be generated. When exporting to another CDF, a new compression and sparseness can be specified for each variable. When exporting to the terminal screen or a text file, the format of the output can be tailored as necessary.

3.3.2 Special Attribute Usage

CDFexport uses the following special attributes:

`FORMAT`	Used as the initial value in a variable's `Format` field.
`VALIDMIN`	Used as the initial filter value in a variable's `Minimum` field.
`VALIDMAX`	Used as the initial filter value in a variable's `Maximum` field.
`FILLVAL`	Used as the initial value in a variable's `FillValue` field.
`MONOTON`	Used as the initial setting in a variable's `Monotonicity` field.

These fields are described in the online help for the appropriate menu. The values of these fields can be changed at any time. The special attributes are simply used to provide initial values. Note also that the usage of these special attributes can be controlled by the options selected with the "initial" qualifier.

3.3.3 Executing the CDFexport Program

Usage:

VMS:

```
$ CDFEXPORT [/INITIAL=(<options>)] [/[NO]PROMPT] [/ZMODE=<mode>]
            [/REPORT=(<types>)] [/[NO]STATISTICS] [/[NO]NEG2POSFP0]
            [/CACHE=(<sizes>)] [/[NO]SIMPLE] [/BATCH=<mode>]
            [/CDF=<path>] [/TEXT=<path>] [/SETTINGS=<path>] <cdf-spec>
```

UNIX:

```
% cdfexport [-initial "<options>"]  [-[no]prompt] [-zmode <mode>]
            [-report "<types>"] [-[no]statistics] [-[no]neg2posfp0]
            [-cache "<sizes>"] [-[no]simple] [-batch <mode>]
            [-cdf <path>] [-text <path>] [-settings <path>] <cdf-spec>
```

MS-DOS:[6]

```
> cdfexport [-initial "<options>"]  [-[no]prompt] [-zmode <mode>]
            [-report "<types>"] [-[no]statistics] [-[no]neg2posfp0]
            [-cache "<sizes>"] [-[no]simple] [-batch <mode>]
            [-cdf <path>] [-text <path>] [-settings <path>] <cdf-spec>
```

Macintosh:

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

Parameter(s):

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

The specification of the CDF(s) from which to export. 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.

VMS, UNIX, & MS-DOS: If the "prompt" qualifier is used, this will appear as the initial specification. If this parameter is omitted, the "prompt" qualifier must be specified and the initial specification will be the current directory.

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.

Qualifier(s):

`/[NO]PROMPT` (VMS)
`-[no]prompt` (UNIX & MS-DOS)

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

Not applicable. (Macintosh)

Specifies whether or not a prompt is issued for the CDF(s) specification. If this qualifier is not specified, the CDF(s) specification must be entered on the command line and is automatically opened.

VMS, UNIX, & MS-DOS: If a CDF(s) specification was entered on the command line, that CDF(s) specification will initially appear at the prompt. Otherwise, the current directory will appear at the prompt.

Macintosh: This qualifier is not applicable on the Macintosh. The CDF(s) specification must always be entered in the dialog box and is then automatically opened.

`/INITIAL=(<defaults>)` (VMS)
`-initial "<defaults>"` (UNIX & MS-DOS)
`Initial options` check boxes, see below (Macintosh)

The default settings that are initially in affect when a CDF is opened. These setting are only the settings initially in effect. The user may change any of them at any time. More detailed descriptions of each option may be found in the appropriate sections that follow.

VMS, UNIX, & MS-DOS: `<defaults>` is a comma-separated list of settings consisting of one or more of the options in the list that follows. **Macintosh:** The initial settings are selected using the check boxes described in the list that follows.

`[NO]FILTER` (VMS)
`[no]filter` (UNIX & MS-DOS)
`Filters enabled` check box (Macintosh)

Whether or not each item/variable is initially filtered.

`[NO]FILLS` (VMS)
`[no]fills` (UNIX & MS-DOS)
`Use fills` check box (Macintosh)

Whether or not the use of fill values is enabled.

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

Specifies whether or not a variable's `FORMAT` attribute entry is used as its initial "format" field.

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

Specifies whether or not a variable's `FILLVAL` attribute entry is used as its initial "fill value" field.

`[NO]VALIDMIN` (VMS)
`[no]validmin` (UNIX & MS-DOS)
`Use VALIDMIN` check box (Macintosh)

Specifies whether or not a variable's `VALIDMIN` attribute entry is used as its initial minimum filter value.

`[NO]VALIDMAX` (VMS)

`[no]validmax` (UNIX & MS-DOS)
`Use VALIDMAX` check box (Macintosh)

Specifies whether or not a variable's `VALIDMAX` attribute entry is used as its initial maximum filter value.

`[NO]MONOTON` (VMS)
`[no]monoton` (UNIX & MS-DOS)
`Use MONOTON` check box (Macintosh)

Specifies whether or not a variable's `MONOTON` attribute entry is used as its initial monotonicity.

`[NO]RECORD` (VMS)
`[no]record` (UNIX & MS-DOS)
`Show 'Record'` check box (Macintosh)

Specifies whether or not the `Record` item will be present.

`[NO]INDICES` (VMS)
`[no]indices` (UNIX & MS-DOS)
`Show 'Indices'` check box (Macintosh)

Specifies whether or not the `Indices` item will be present.

`[NO]EXCLUSIVE` (VMS)
`[no]exclusive` (UNIX & MS-DOS)
`Exclusive filters` check box (Macintosh)

Specifies whether or not exclusive filters are allowed.

`[NO]OUTPUT` (VMS)
`[no]output` (UNIX & MS-DOS)
`Outputs enabled` check box (Macintosh)

Specifies whether or not each item/variable is initially output.

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

Specifies the initial setting of whether or not an existing CDF will be deleted when a new CDF is created with the same name.

`[NO]PREALLOCATE` (VMS)
`[no]preallocate` (UNIX & MS-DOS)
`Preallocate records` check box (Macintosh)

Specifies the initial setting of whether or not variable records are to be preallocated when creating a new CDF.

`SINGLE` or `MULTI` (VMS)
`single` or `multi` (UNIX & MS-DOS)
`New format` radio buttons (Macintosh)

Specifies the initial setting of whether single-file or multi-file CDFs are created.

`HOST` or `NETWORK` (VMS)
`host` or `network` (UNIX & MS-DOS)
`New encoding` radio buttons (Macintosh)

Specifies the initial setting of whether host-encoded or network-encoded CDFs are created.

`ROW` or `COLUMN` (VMS)
`row` or `column` (UNIX & MS-DOS)
`Majority` radio buttons (Macintosh)

Specifies the initial setting of whether row-major, column-major, or input-major CDFs/listings are generated. Input-majority is the majority of the input CDF. **VMS, UNIX, & MS-DOS:** Input-majority is selected by specifying neither row-majority nor column-majority.

`EPOCH`, `EPOCH1`, `EPOCH2`, `EPOCH3`, `EPOCHf`, or `EPOCHx` (VMS)
`epoch`, `epoch1`, `epoch2`, `epoch3`, `epochf`, or `epochx` (UNIX & MS-DOS)
`EPOCH style` radio buttons (Macintosh)

Specifies the initial EPOCH encoding style.

`HORIZONTAL` or `VERTICAL` (VMS)
`horizontal` or `vertical` (UNIX & MS-DOS)
`Orientation` radio buttons (Macintosh)

Specifies the initial setting of whether horizontal or vertical listings are generated.

Note that these options can be changed at any time after the CDF has been opened. If this qualifier is not specified, each of these options has a default setting. These default settings are also used for options not specified with this qualifier.

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

Specifies which zMode should be used. 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 encountered in 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. The `<types>` option 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=(<sizes>)` (VMS)

`-cache "<sizes>"` (UNIX & MS-DOS)
`Cache sizes` edit field (Macintosh)

Specifies the cache sizes to be used by the CDF library for the dotCDF file and the various scratch files. The `<sizes>` option is a comma-separated list of `<size><type>` pairs where `<size>` is a cache size and `<type>` is the type of file as follows: `d` for the dotCDF file, `s` for the staging scratch file, and `c` for the compression scratch file. For example, `200d,100s` specifies a cache size of `200` for the dotCDF file and a cache size of `100` for the staging scratch file. The dotCDF file cache size can also be specified without the `d` file type for compatibility with older CDF releases (e.g., `200,100s`). Note that not all of the file types must be specified. Those not specified will receive a default cache size chosen by the CDF library. A cache size is the number of 512-byte buffers to be used. Section 2.1.5 explains the caching scheme used by the CDF library.

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

Specifies whether or not caching statistics are displayed when a CDF is closed.

`/[NO]SIMPLE` (VMS)
`-[no]simple` (UNIX & MS-DOS)
`Not applicable.` (Macintosh)

Specifies if a simplified version of CDFexport should be executed. The following conditions apply to simple mode:

- Only text listings can be generated (to the screen or a file).
- No filtering is available.
- When listing to a text file, FORMAT attribute entries are ignored and standard formats are used instead.
- Only a limited set of the options for the 'initial' qualifier may be specified.
- zMode/2 is used by default.
- Horizontal listings are created by default.

`/BATCH=<mode>` (VMS)
`-batch <mode>` (UNIX & MS-DOS)
`Not applicable.` (Macintosh)

Specifies if CDFexport should execute in a non-interactive batch mode. The mode option may be either "text" to generate a text file listing or "cdf" to output to a new CDF. A settings file will be used if one exists with the default name in the current directory or is explicitly specified with the 'settings' qualifier. The settings file contains the parameters necessary to specify how the output CDF or text file should be generated. If a settings file is not available, default parameters will be used. CDFexport must be used interactively to create a settings file.

`/CDF=<cdf>` (VMS)
`-cdf <cdf>` (UNIX & MS-DOS)
`Not applicable.` (Macintosh)

Specifies an output CDF pathname to be used when exporting to a CDF in batch mode. Do not include an extension. When executing interactively this pathname will initially appear at the output CDF prompt. If this qualifier is not specified, the default CDF name is "default" (in the current directory).

`/TEXT=<path>` (VMS)
`-text <path>` (UNIX & MS-DOS)
`Not applicable.` (Macintosh)

Specifies a pathname to be used when exporting to a text file listing in batch mode. When executing interactively this pathname will initially appear at the text file prompt. If this qualifier is not specified, the default text file name is "default.lis" (in the current directory).

`/SETTINGS=<path>` (VMS)
`-settings <path>` (UNIX & MS-DOS)
`Not applicable.` (Macintosh)

Specifies a settings file pathname to be used when executing in batch mode. When executing interactively this pathname will initially appear at the settings file prompt when saving/restoring the current settings. The default settings file is "simple.set" if executing in simple mode and "export.set" otherwise (with each being in the current directory).

Example(s):

VMS:

```
$ CDFEXPORT [.SAMPLES]
$ CDFEXPORT/ZMODE=2/CACHE=(50d,100s) GISS_WETLX
$ CDFEXPORT/PROMPT/REPORT=(W,E)/INITIAL=(EXCLUSIVE,NOFORMAT)
$ CDFEXPORT/SIMPLE/BATCH=TEXT/TEXT=FLUX.OUT FLUX1996
```

UNIX:

```
% cdfexport samples
% cdfexport -zmode 2 -cache "50d,100s" giss_wetl
% cdfexport -prompt -report "w,e" -initial "exclusive,noformat"
% cdfexport -simple -batch text -text flux.out flux1996
```

MS-DOS:

```
> cdfexport samples
> cdfexport -zmode 2 -cache "50d,100s" giss_wetl
> cdfexport -prompt -report "w,e" -initial "exclusive,noformat"
```

3.3.4 Interaction with CDFexport

Interaction with CDFexport is through a 4-part `SelectionWindow`, an `ActionMenu`, an `OptionMenu`, numerous prompt windows, and several screen listing windows. Detailed online help is available for each window so only a brief description of each will be given here. After selecting a CDF from which to export, part 1 of the `SelectionWindow` will be loaded with a line for the `<Record>` item, the `<Indices>` item, and each variable.

The <Record> item allows the record number to be included in a screen/file listing and/or filtering on the record number for any type of output. The <Indices> item allows the dimension indices to be included in a screen/file listing and/or filtering on the dimension indices for any type of output. Each variable line allows that variable to be included and/or filtered when generating any type of output. The KeyDefinitions window displays the available functions and their corresponding keys for a given window/prompt. The MessageBuffer displays errors/instructions as necessary.

Cycling through the four parts of the SelectionWindow allows the selection of the output to be generated. The online help explains the purpose of each field in the four parts of the SelectionWindow. The OptionMenu allows additional selections affecting the output. The ActionMenu is then used to generate the desired type of output (as well as some other miscellaneous operations).

The easiest way to learn how to use CDFexport is to read through the online help while generating the various types of output using a CDF with which you are familiar.

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. The format of the CDF may be changed (see Section 2.2.7).
2. The data encoding of the CDF may be changed (see Section 2.2.8).
3. The variable majority of the CDF may be changed (see Section 2.3.15).
4. The compression of the CDF (see Section 2.2.10) or the CDF's variables (see Section 2.3.14) may be changed.
5. The sparseness of the CDF's variables may be changed (see Sections 2.3.12 and 2.3.13).

3.4.2 Executing the CDFconvert Program

Usage:

VMS:

```
$ CDFCONVERT [/SKELETON=<skt-cdf-path>] [/[NO]LOG]
             [/[NO]PERCENT] [/REPORT=(<types>)]
             [/CACHE=(<sizes>)] [/[NO]PAGE] [/[NO]STATISTICS]
             <src-cdf-spec>
                [/ZMODE=<mode>] [/[NO]NEG2POSFP0]
             <dst-cdf-spec>
```

```
                 [/SINGLE | /MULTI] [/ROW | /COLUMN] [/[NO]DELETE]
                 [/ENCODING=<encoding> | /HOST | /NETWORK]
                 [/COMPRESSION=(<types>)] [/SPARSENESS=(<types>)]
```

UNIX:

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

MS-DOS:[7]

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

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.

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)

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

`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. A prompt for the `RETURN` key will be issued after each page. A page is generally 22 lines of output.

`/[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.)

`/COMPRESSION=(<types>)` (VMS)
`-compression <types>` (UNIX & MS-DOS)
`Compression` edit field (Macintosh)

Specifies the types of compression to be used for the CDF and/or variables. The <types> option consists of a comma-separated list of the following...

cdf:<cT>	CDF's compression.
vars:<cT>	Compression for all variables.
vars:<cT>:<bF>	Compression for all variables with a blocking factor specified.
vars:<cT>:<bF>:<r%>	Compression for all variables with a blocking factor and reserve percentage specified.
var:<name>:<cT>	Compression for one particular variable.
var:<name>:<cT>:<bF>	Compression for one particular variable with a blocking factor specified.
var:<name>:<cT>:<bF>:<r%>	Compression for one particular variable with a blocking factor and reserve percentage specified.

Where <cT> is one of the following compressions: none, rle.0, huff.0, ahuff.0, or gzip.<level>; <bF> is a blocking factor; <r%> is a reserve percentage; and <name> is a delimited, case-sensitive variable name with the following syntax:

<delim><char1><char2>...<charN><delim>

In general, do not use single or double quote marks as delimiters. **VMS:** The entire delimited variable name must be enclosed in double quote marks (to preserve case-sensitivity).

For the gzip compression, <level> must be in the range from 1 (fastest compression) to 9 (best compression).

For compressions not specified the compression in the source CDF will be used. Specifying a variable compression using var:... overrides a compression specified with vars:...

/SPARSENESS=(<types>) (VMS)
-sparseness <types> (UNIX & MS-DOS)
Sparseness edit field (Macintosh)

Specifies the types of sparseness to be used for the variables. The <types> option consists of a comma-separated list of the following...

vars:<sT>	Sparseness for all variables.
var:<name>:<sT>	Sparseness for one particular variable.

Where <sT> is one of the following: srecords.no, srecords.pad, or srecords.prev; and <name> is a delimited, case-sensitive variable name with the following syntax:

<delim><char1><char2>...<charN><delim>

In general, do not use single or double quote marks as delimiters. **VMS:** The entire delimited variable name must be enclosed in double quote marks (to preserve case-sensitivity).

For sparsenesses not specified the sparseness in the source CDF will be used. Specifying a variable sparseness using var:... overrides a sparseness specified with vars:...

`/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 encountered in 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. The `<types>` option 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=(<sizes>)` (VMS)
`-cache "<sizes>"` (UNIX & MS-DOS)
`Cache sizes` edit field (Macintosh)

Specifies the cache sizes to be used by the CDF library for the dotCDF file and the various scratch files. The `<sizes>` option is a comma-separated list of `<size><type>` pairs where `<size>` is a cache size and `<type>` is the type of file as follows: `d` for the dotCDF file, `s` for the staging scratch file, and `c` for the compression scratch file. For example, `200d,100s` specifies a cache size of `200` for the dotCDF file and a cache size of `100` for the staging scratch file. The dotCDF file cache size can also be specified without the `d` file type for compatibility with older CDF releases (e.g., `200,100s`). Note that not all of the file types must be specified. Those not specified will receive a default cache size chosen by the CDF library. A cache size is the number of 512-byte buffers to be used. Section 2.1.5 explains the caching scheme used by the CDF library.

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

Specifies whether or not caching statistics are displayed when a CDF is closed.

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 CDFcompare

3.5.1 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).

3.5.2 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=(<sizes>)]
             [/[NO]PAGE] [/[NO]STATISTICS] [/[NO]PERCENT] [/[NO]VALUE]
             <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 "<sizes>"]
             [-[no]page] [-[no]statistics] [-[no]percent] [-[no]value]
             <cdf-spec-1> <cdf-spec-2>
```

MS-DOS:[8]

```
> cdfcompare [-[no]log] [-[no]attr] [-[no]var] [-[no]number]
             [-[no]etc] [-[no]neg2posfp0] [-zmodes "<mode1>,<mode2>"]
             [-[no]location] [-report "<types>"] [-cache "<sizes>"]
             [-[no]page] [-[no]statistics] [-[no]percent] [-[no]value]
             <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

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

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.

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 encountered in 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. A prompt for the `RETURN` key will be issued after each page. A page is generally 22 lines of output.

`/[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>]
```

`/[NO]VALUE` (VMS)
`-[no]value` (UNIX & MS-DOS)
`Display values` check box (Macintosh)

Specifies whether or not the values are displayed when a difference is detected between variable values or attribute entries. Note that for variable values to be displayed, the display of the locations of the differences must also be enabled.

`/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. The `<types>` option 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=(<sizes>)` (VMS)
`-cache "<sizes>"` (UNIX & MS-DOS)
`Cache sizes` edit field (Macintosh)

Specifies the cache sizes to be used by the CDF library for the dotCDF file and the various scratch files. The `<sizes>` option is a comma-separated list of `<size><type>` pairs where `<size>` is a cache size and `<type>` is the type of file as follows: `d` for the dotCDF file, `s` for the staging scratch file, and `c` for the compression scratch file. For example, `200d,100s` specifies a cache size of `200` for the dotCDF file and a cache size of `100` for the staging scratch file. The dotCDF file cache size can also be specified without the `d` file type for compatibility with older CDF releases (e.g., `200,100s`). Note that not all of the file types must be specified. Those not specified will receive a default cache size chosen by the CDF library. A cache size is the number of 512-byte buffers to be used. Section 2.1.5 explains the caching scheme used by the CDF library.

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

Specifies whether or not caching statistics are displayed when a CDF is closed.

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.

3.5.3 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.

3.6 CDFstats

3.6.1 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 a valid range of values (with illegal/fill values being ignored), and the variable's monotonicity.

Monotonicity refers to whether or not a variable's 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.

rVariable	Record Variance	Dimension Variances	Check Monotonicity?
`EPOCH`	`VARY`	`NOVARY,NOVARY`	Yes
`LATITUDE`	`NOVARY`	`VARY,NOVARY`	Yes
`LONGITUDE`	`NOVARY`	`NOVARY,VARY`	Yes
`ELEVATION`	`NOVARY`	`VARY,VARY`	No
`TEMPERATURE`	`VARY`	`VARY,VARY`	No

Table 3.1: Example rVariables, CDFstats Monotonicity Checking

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

3.6.2 Special Attribute Usage

CDFstats uses the following special attributes:

`FORMAT` Used when displaying a variable statistic (e.g., minimum variable value).

`VALIDMIN` If range checking is enabled, used as the minimum valid value for a variable. For a variable with a non-character data type, only the first element of its `VALIDMIN` attribute entry is used. Also, if requested, the `VALIDMIN` attribute entry for a variable will be updated with the actual minimum value found. Again, if the variable has a non-character data type the `VALIDMIN` attribute entry will be updated to have just one element.

`VALIDMAX` If range checking is enabled, used as the maximum valid value for a variable. For a variable with a non-character data type, only the first element of its `VALIDMAX` attribute entry is used. Also, if requested, the `VALIDMAX` attribute entry for a variable will be updated with the actual maximum value found. Again, if the variable has a non-character data type the `VALIDMAX` attribute entry will be updated to have just one element.

`FILLVAL` If fill value usage is enabled, used as the value which is ignored while collecting statistics for a variable.

`MONOTON` If requested, the `MONOTON` attribute entry for a variable will be updated with the actual monotonicity found. The possible values for the `MONOTON` attribute entry are described in Section 3.1.3.

`SCALEMIN` If requested, the `SCALEMIN` attribute entry for a variable will be updated with the actual minimum value found.

`SCALEMAX` If requested, the `SCALEMAX` attribute entry for a variable will be updated with the actual maximum value found.

The usage of these special attributes can be controlled with command line qualifiers.

3.6.3 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=(<sizes>)] [/[NO]STATISTICS]
           <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 "<sizes>"] [-[no]statistics]
           <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 "<sizes>"] [-[no]statistics]
           <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. Note that for variables having a non-character data type only the first element of the `VALIDMIN` and `VALIDMAX` attribute entries are used.

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

Specifies whether or not fill values are ignored when collecting statistics. The `FILLVAL` attribute entry for a variable (if it exists) is used as the fill value.

`/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 at a time. A prompt for the `RETURN` key will be issued after each page. A page is generally 22 lines of output.

`/[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 encountered in 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. The <types> option 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=(<sizes>) (VMS)
-cache "<sizes>" (UNIX & MS-DOS)
Cache sizes edit field (Macintosh)

Specifies the cache sizes to be used by the CDF library for the dotCDF file and the various scratch files. The <sizes> option is a comma-separated list of <size><type> pairs where <size> is a cache size and <type> is the type of file as follows: d for the dotCDF file, s for the staging scratch file, and c for the compression scratch file. For example, 200d,100s specifies a cache size of 200 for the dotCDF file and a cache size of 100 for the staging scratch file. The dotCDF file cache size can also be specified without the d file type for compatibility with older CDF releases (e.g., 200,100s). Note that not all of the file types must be specified. Those not specified will receive a default cache size chosen by the CDF library. A cache size is the number of 512-byte buffers to be used. Section 2.1.5 explains the caching scheme used by the CDF library.

/[NO]STATISTICS (VMS)
-[no]statistics (UNIX & MS-DOS)

`Display statistics` check box (Macintosh)

Specifies whether or not caching statistics are displayed when a CDF is closed.

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.

3.6.4 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").

3.7 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 to 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.

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 Special Attribute Usage

The special attribute `FORMAT` is used by SkeletonTable (depending on the setting of the "format" qualifier) when writing variable values in a skeleton table.

3.7.3 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=(<sizes>)] [/[NO]PAGE] [/[NO]STATISTICS]
                <cdf-path>
```

UNIX:

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

MS-DOS:[9]

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

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