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):

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

The default settings that initially appear on the Input Options, Output Options, and Listing Options windows. Whether or not to automatically select zMode may also be specified. These settings 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.

ZMODE (VMS)
zmode (UNIX & MS-DOS)
zMode check box (Macintosh)

When specified, zMode is automatically selected when a CDF is opened.

[NO]ERASE (VMS)
[no]erase (UNIX & MS-DOS)
Erase, new CDF check box (Macintosh)

Specifies whether or not to erase the Variable Selection window when a new CDF is selected.

[NO]FILTER (VMS)
[no]filter (UNIX & MS-DOS)
Use filters check box (Macintosh)

Specifies whether or not to use the variable filters when generating output.

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

Specifies whether or not to use the FILLVAL attribute when filtering.

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

Specifies whether or not to use the FORMAT attribute when encoding values to be displayed.

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

Specifies whether or not to convert -0.0 to 0.0 when read from or written to a CDF. -0.0 is an illegal floating-point value on VAXes and DEC Alphas running

OpenVMS.

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

Specifies the format of created CDFs.

`HOST | NETWORK` (VMS)
`host | network` (UNIX & MS-DOS)
`Host/Network` radio buttons (Macintosh)

Specifies the encoding of created CDFs.

`HORIZONTAL | VERTICAL` (VMS)
`horizontal | vertical` (UNIX & MS-DOS)
`Horizontal/Vertical` radio buttons (Macintosh)

Specifies the type of text output desired while in zMode.

`EPOCH | EPOCH1` (VMS)
`epoch | epoch1` (UNIX & MS-DOS)
`Epoch/Epoch1` radio buttons (Macintosh)

Specifies the style in which `CDF_EPOCH` values should be displayed.

`/SPECIFICATION=<default-cdf-spec>` (VMS)
`-specification <default-cdf-spec>` (UNIX & MS-DOS)
Not supported (Macintosh)

VMS, UNIX, & MS-DOS: A default CDF(s) specification for the CDF name field. The user may choose this specification or enter another. This qualifier may not be specified if the initial CDF(s) specification parameter was specified.

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

Specifies the cache size to be used for a 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:

```
$ CDFLIST
$ CDFLIST GISS_WETL
$ CDFLIST NCDS$DATA:FGGE3B
$ CDFLIST/INITIAL=(ZMODE,HORIZONTAL)/SPECIFICATION=NCDS$DATA:
```

UNIX:

```
% cdflist
```

```
% cdflist ../cac_sst_blended
% cdflist ~/CDFs
% cdflist -initial "zmode,horizontal" -specification ~/CDFs
```

MS-DOS:

```
> cdflist
> cdflist ..\cac_sst
> cdflist a:\cdfs
> cdflist -initial "zmode,horizontal" -specification a:\cdfs
```

3.10.3 Interaction with CDFlist

The user interface to `CDFlist` is contained on a single screen divided into several fixed windows and pop-up windows. The `Action Menu` controls the execution of the program. This menu allows you to move to a different window, start the data output, browse a help file, or exit the program. You must return to this menu each time you want to select a new variable, output option, listing type, or input CDF. This menu may be displayed at any time by entering the `Action` key. Use the `Up` and `Down` keys to highlight a choice in the `Action Menu` and then enter the `Select` key to execute that function. The `Action Menu` appears in Figure 3.1.

```
Generate Output
Select Input CDF name
Select Output Device
Select Variables
Select Monotonic Variables
Save Variables to file
Restore Variables from file
Select r/z/zMODE Variables
Select CDF Input Options
Select CDF Output Options
Select CDF Listing Options
Help
Exit
```

Figure 3.1: `Action Menu`, `CDFlist`

Two very important windows are the `Keypad Definition` and `Message` windows. They contain information on keyboard functionality (i.e., which keys to use to perform the operations described below) and program status/error messages.

When `CDFlist` is executed (and a CDF and/or directory is not entered on the command line), you must first specify the CDF you wish to list. If you enter a disk/directory/wildcard specification (without a CDF name), all of the CDFs meeting the specification will appear in a separate pop-up window. Use the `Up` and `Down` keys to choose a CDF and then enter the `Select` key to select that CDF.

The `Output` window will then be displayed with the valid choices. Use the `Up` and `Down` keys to move up and down within the window and then enter the `Select` key to select the desired output option. If output to an

ASCII text file or CDF is selected, you will be prompted for a file/CDF name. You may enter any file/CDF name valid for the operating system being used or accept the default file/CDF name that appears. Do not enter a file extension. (It is automatically appended.) The file extension for a generated ASCII text file is `.lis`.

The second window (that remains constant on the screen) is the `Variable Selection` window. This window allows the user to select the variables to be listed or put into a CDF along with any desired filters on the variables. You will automatically be placed in this window after you have entered an output option. You can also enter this window by requesting the `Action Menu` and then selecting the `Select Variables` option.

There are four fields on each line in the `Variable Selection` window: variable name, minimum filter value, maximum filter value, and output toggle. You can move forward to the next field by entering the `Next Field` key or backwards to the previous field with the `Prev Field` key. You may also use the `Up` and `Down` keys to move vertically through the fields. When the cursor is positioned in a variable name field, entering the `Pop-up` key will display the variables in the CDF in a pop-up window. Use the arrow keys to highlight a variable and then enter the `Select` key to select it. You can quit the pop-up window without selecting a variable by entering the `Quit` key. You may also use the `Page Up` and `Page Down` keys to traverse through the pop-up window. The first line in this pop-up menu contains `**SELECT ALL**`. Selecting this item causes all of the variables to be selected. The second line in this pop-up window contains `**RECORD NUM**`. This simply gives the user the opportunity to display/filter on the record numbers in the CDF.

After selecting a variable, the cursor is positioned in the minimum filter field. A filter is simply the range of data that you would like to see from the selected variable. If the CDF contains a `VALIDMIN` and `VALIDMAX` attribute entry for the selected variable, these values will be displayed as default minimum/maximum filter values. If just specifying a minimum filter value is sufficient, a maximum filter value need not be entered. You do not have to enter a filter for each selected variable. For a line of output to be written for a given record/indices, each filter must be passed. There are two ways to enter minimum and maximum filters. The first is by simply entering the desired values in the appropriate fields. The second method can only be used for variables that consist of a discrete set of values. This method involves entering the `Pop-up` key (when in the desired filter field), which displays a pop-up window that contains the discrete values for the variable. The desired filter value may then be selected. (The `Page Up` and `Page Down` keys may also be used to traverse through the window.)

The minimum/maximum fields (filters) will initially be highlighted. This indicates that the corresponding filter will be used for a variable (if filter values are present). A filter for a particular variable may also be turned off (without erasing the filter values) by entering the `Filter On/Off` key when on the variable name field. This will toggle off the highlighting of the filter values and indicates that no filtering is to be performed on that variable. Entering the `Filter On/Off` key again toggles filtering back on for that variable. Note that erasing the filter values is another way to turn off filtering for a variable.

The output toggle field simply gives the user the option not to view a variable they have selected. A user may only wish to filter on a variable and not view it. When positioned on this field, the user simply enters the `Out on` key to display the variable and `Out Off` key to not display the variable. Note that the user may also enter `Out On/Off` keys from the variable name field.

When all of the desired variables and filters have been selected, enter the `Action` key to display the `Action Menu`. Selecting `Generate Output` will cause the listing or new CDF to be generated. Note that the amount of time needed to generate the output will in most cases be significantly reduced if the `MONOTON` attribute and corresponding entries are present in the CDF (and the variables being filtered are monotonic). See the section explaining `Specifying Monotonic Variables` for more information on monotonic variables. When `CDFlist` completes the generation of the output, the `Action Menu` is redisplayed. It may then be used to select a new CDF to list, respecify the output option, select new variables/filters, get on-line help, select special options, or exit the program.

MS-DOS: A listing may be terminated early by entering the `Quit` key.

When on the `Action Menu` options are available to save/restore the current selections in the `Variable Selection` window. A user may wish to exit the program and upon returning restore the current selections quickly to the window. When the `Save Variables To File` option is selected, the user is prompted for a file name. (Do not enter an extension.) All of the selections in the `Variable Selection` window are stored in that file. A default file name of `cdflist` is available by just entering the `Select` key at the prompt. When the `Restore Variables From File` option is selected, the user is again prompted for a file name. (`cdflist` is again the default.) The selections are restored to the `Variable Selection` window from that file. `CDFlist` will check the current CDF for the existence of each variable (displaying only those variables that do exist).

Another option on the `Action Menu` is `Select r/z/zMODE Variables`. `CDFlist` groups variables in three separate manners: rVariables, zVariables, and zMode. All of the rVariables are contained in one group; zVariables are grouped together by dimensionality; and zMode consists of all of the variables in the CDF with the rVariables being made to appear as zVariables. Variables from only one of these groups at a time may be selected for filtering/output. See Section 2.3 for more information on r/zVariables and Section 2.1.2 for more information on zMode.

On the `Action Menu` three windows of special options are available. Each option brings up a window containing a set of small option selection lists (an example of which is shown in Figure 3.2). The user makes selections by moving the `*` (the current choice) with the `Up` and `Down` keys and moving to the next list by entering the `Select` key. When entering an option selection list, if the current choice is the one desired, just enter the `Select` key to move to the next list. To return back to the `Action Menu`, enter the `Quit` key at any time.

```
(*) Single
( ) Multi
```

Figure 3.2: Option Selection List, `CDFlist`

Selecting `Select CDF Input Options` causes a window to be displayed with the following option selection lists:

1. Whether or not to erase the variables in the `Variable Selection` window each time a new input CDF is selected. If users select a CDF that contains duplicate variables, they may wish to reuse the current selections. If users choose not to erase selected variables, `CDFlist` will check to make sure that the current variables actually exist when a new CDF is selected.

Selecting `Select CDF Output Options` causes a window to be displayed with the following option selection lists:

1. When selecting output to a CDF, whether the file format of the new CDF is single-file or multi-file.
2. When selecting output to a CDF, whether the encoding of the new CDF is network or host.

Selecting `Select CDF Listing Options` causes a window to be displayed with the following option selection lists:

1. The format used when displaying EPOCH values may be selected. (The `FORMAT` attribute is always ignored when displaying EPOCH values.)

2. Whether or not to use the `FORMAT` attribute when displaying values may be selected. (The `FORMAT` attribute and corresponding entries must exist in the CDF.)

3. Whether or not to use the `FILLVAL` attribute when filtering (assuming the `FILLVAL` attribute exists). If the `FILLVAL` attribute is to be used, fill values in the data are ignored when determining whether or not to filter out a set (line) of values (i.e., fill values outside the filter range will appear in the output).

4. Whether or not to ignore all of the filters when generating the output. This is a more convenient way to turn off filtering than to do so manually for each variable on the `Variable Selection` window.

5. Whether or not to convert `-0.0` to `0.0` when read from or written to a CDF. `-0.0` is an illegal floating-point value on VAXes and DEC Alphas running OpenVMS. (`CDFlist` may abort with a system error.)

6. Whether to display values vertically or horizontally while in zMode.

On the `Action Menu` the option `Specify Monotonic Variables` gives the user the opportunity to specify that a variable is monotonic. This is useful when a CDF does not contain the attribute `MONOTON` or has the entries incorrectly specified. When this option is executed, a pop-up window is displayed containing all of the variables in the CDF and their monotonicities. If a `MONOTON` attribute entry exists for a variable, it is set accordingly. An `I` is displayed if the values increase, `D` if the values decrease, and `F` if the variable is not monotonic (or if an entry does not exist). If a variable varies with more than one "dimension", `N/A` is displayed — these variables can never be specified monotonic. Once inside the pop-up window, the user traverses through the window as in any other pop-up window but with one exception — the cursor is positioned to the monotonic value of the variable. The user then may change that value by entering the `INCREASE` key for increasing values, the `DECREASE` key for decreasing values, or the `FALSE` key for not monotonic.

3.11 CDFwalk

3.11.1 Introduction

The `CDFwalk` program allows direct access to all variable data in a CDF. The data values of up to 27 variables may be displayed at once. (The CDF can have more variables but only 27 may be viewed at any one time.) The data value for each variable at a given record/indices location within the CDF conceptual view is displayed. The user can "walk" from one record/indices location to another sequentially (in any "direction") or a particular record/indices location can be "jumped" to directly. Variable values may also be used to directly "jump" to a record/indices location.

3.11.2 Executing the CDFwalk Program

Usage:

VMS:

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

UNIX:

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

MS-DOS:

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

Macintosh:

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

Parameter(s):

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

The initial specification of the CDF(s) at which to begin "walking." (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.

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):

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

The default settings that initially appear on the `Special Options` window. 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]FORMAT` (VMS)

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

Specifies whether or not to use the FORMAT when encoding values to be displayed.

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

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

/SPECIFICATION=<default-cdf-spec> (VMS)
-specification <default-cdf-spec> (UNIX & MS-DOS)
Not supported (Macintosh)

A default CDF specification for the CDF name field. The user may choose this specification or enter another. This qualifier may not be specified if the initial CDF(s) specification parameter was specified.

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

Specifies the cache size to be used for a 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:

```
$ CDFWALK
$ CDFWALK [-.TEMP]NOAA7
$ CDFWALK/INITIAL=(NOFORMAT,NEG2POSFP0)/SPECIFICATION=NCDS$DATA:
```

UNIX:

```
% cdfwalk
% cdfwalk ../cac_sst
% cdfwalk /disk3/CDFs/ssclimate
% cdfwalk -initial "noformat,neg2posfp0" -specification ~/CDFs
```

MS-DOS:

```
> cdfwalk
> cdfwalk ..\cac_sst
> cdfwalk c:\cdfs
> cdfwalk -initial "noformat,neg2posfp0" -specification c:\cdfs
```

3.11.3 Interaction with CDFwalk

The user interface to `CDFwalk` is contained on a single screen divided into several fixed windows and pop-up windows. The `Action Menu` controls the execution of the program. This menu allows you to "walk" through a CDF, select a new CDF through which to "walk", select r/zVariables, select `CDFwalk` special options, browse on-line help, or exit the program. This menu may be displayed at any time by entering the `Action` key. Use the `Up` and `Down` keys to highlight a choice in the `Action Menu`, and then enter the `Select` key to execute that function. The `Action Menu` appears in Figure 3.3.

```
Walk Thru CDF data
Select Input CDF name
Select Special Options
Select r/z Variables
Help
Exit
```

Figure 3.3: `Action Menu`, `CDFwalk`

Two very important windows are the `Keypad Definition` and `Message` windows. They contain information on keyboard functionality (i.e., which keys to use to perform the operations described below) and program status/error messages.

When `CDFwalk` is executed (and a CDF and/or directory/wildcard specification was not entered on the command line), you must first specify the CDF through which you wish to "walk." If you enter a directory/wildcard specification, all of the CDFs meeting that specification will appear in a separate pop-up window. Use the `Up` and `Down` keys to choose a CDF and then enter the `Select` key to "walk" through that CDF.

After the input CDF is selected, the `Record/Indices` window appears. The size of this window depends on the dimensionality of the variables being displayed (initially the rVariables if any exist). Vertically, the window will contain one row for the record number and one row for each dimension of the variables. Horizontally, each line of the window contains three fields. The first field contains a variable name; the second field contains a data value for the variable; and the third field contains a record number or dimension index.

The variable selected for the record number line will be one that only varies from record to record (not varying along any of the dimensions). If more than one variable of that type exists, a pop-up window will be displayed containing the list of variables. Select one of the variables, keeping in mind that you will have an opportunity to use it for positioning by value to a desired record. If a variable meeting the requirements mentioned above does not exist, only the record number field will be used.

The variable selected for each dimension line will be one that only varies along the corresponding dimension (and does not vary from record to record). Again, if more than one variable of that type exists in the CDF, a pop-up window will be displayed to allow you to select the variable to use for that dimension. Keep in mind that you will be able to use it to position by value to a desired index along that dimension. If for a particular dimension a variable meeting the requirements mentioned above does not exist, only the dimension index field will be used.

After the `Record/Indices` window has been specified (which will usually occur automatically), the `Variable Values` window will be displayed. It contains up to 27 variable names and the associated data values for the record/indices shown in the `Record/Indices` window.

Initially, the cursor will be positioned at the data value field of the record number line in the `Record/Indices` window (assuming that a variable was found meeting the above requirements; if not, the cursor will start at the record number field). You may now begin to "walk" through the CDF by using the right arrow to increment to and the left arrow to decrement to a new record. For each new record the data for each variable is updated in both the `Record/Indices` and `Variable Values` windows. If the CDF contains many records, you may want to advance quickly to a desired record. You may do this by entering the `Input` key. `CDFwalk` enters an input mode and prompts for the value to which to advance. If the exact value is not in the CDF for the variable, the record number is changed to correspond to the closest possible value. Another method of advancing to a desired value is by entering the `Pop-up` key. This will cause a pop-up window containing all of the values for the variable to be displayed. You may select the value desired as you would with any other pop-up window and `CDFwalk` will automatically position the record number to the corresponding value.

After positioning to a desired record (and if the CDF is multidimensional) you may now wish to "walk" through the dimension indices. This is done by entering the up and down arrow keys to position the cursor to the desired dimension line. "Walking" through dimensions is done in the same way as "walking" through the record numbers.

You may also wish to "walk" through the CDF by using the actual record number or dimension indices rather than data values. (This would also be necessary if variables meeting the above requirements were not found.) This is done by entering the `Next Field` key, which causes the cursor to advance to the next field (i.e., from the data value field to the record number/indices field). Entering the `Back Field` key causes the cursor to move back to the previous field. "Walking" through the CDF using the record number/indices fields is the same as using the data value fields but with a few exceptions. First, no `Pop-up` option is available with these fields. Also, in input mode (after entering the `Input` key) you may input either an absolute record number/index to which to move or a relative offset from the current record number/index. Prefixing the value with a `+/-` will increment/decrement the record number/index by that amount.

If more than 27 variables exist and a particular variable is not currently displayed in the `Variable Values` window, an option is provided to scroll new variables into that window. This is done by entering the `Scroll` key. A small pop-up window is displayed in which you may use the `Next/Prev Variable` keys to scroll in or scroll out variables. When the desired set of variables is displayed, enter the `Select` key to continue "walking" through the CDF.

When you are finished "walking" through the CDF, enter the `Action` key to display the `Action Menu`. You may input a new CDF, resume "walking" through the current CDF, select special options, read on-line help, or exit the program.

On the `Action Menu` selecting `Special Options` causes a window to be displayed with the following option selection lists:

1. Whether or not to use the `FORMAT` attribute when displaying values may be selected. (The `FORMAT` attribute and corresponding entries must exist in the CDF.)
2. Whether or not to convert `-0.0` to `0.0` when read from a CDF. `-0.0` is an illegal floating-point value on VAXes and DEC Alphas running OpenVMS. (`CDFwalk` may abort with a system error.)

The user makes a selection by moving the `*` (the current choice) with the `Up` and `Down` keys and moving to the next options selection list (if another exists) by entering the `Select` key. An example option selection list is shown in Figure 3.4. To return to the `Action Menu`, enter the `Quit` key at any time.

Another option on the `Action Menu` is `Select r/z Variables`. `CDFwalk` groups variables in two separate

```
(*) CDF FORMAT attribute
( ) CDFWALK defaults
```

Figure 3.4: Option Selection List, `CDFwalk`

manners: rVariables and zVariables. All of the rVariables are in one group and the zVariables are grouped together by dimensionality. Only one of these groups may be "walked" through at a time. For more information on the r/zVariables see Section 2.3.

3.12 CDFedit

3.12.1 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.[12]

3.12.2 Executing the CDFedit Program

Usage:

VMS:

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

UNIX:

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

MS-DOS:

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

[12] 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 read from or written to a CDF. (The CDF is not modified.) `-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 a 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]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 editing session.

`/[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 GISS_WETLX
$ CDFEDIT/BROWSE/PROMPT/REPORT=(ERRORS)
```

UNIX:

```
% cdfedit samples
% cdfedit -zmode 2 -noformat giss_wetl
% cdfedit -browse -prompt -report "errors"
```

MS-DOS:

```
> cdfedit samples
> cdfedit -zmode 2 -noformat giss_wetl
> cdfedit -browse -prompt -report "errors"
```

3.12.3 Interaction with CDFedit

Interaction with `CDFedit` is through a series of menus and windows. Extensive on-line help is provided and will not be repeated here.[13] The on-line help does refer to the sections of a window by name. Figure 3.5 illustrates the various sections of the possible types of windows.

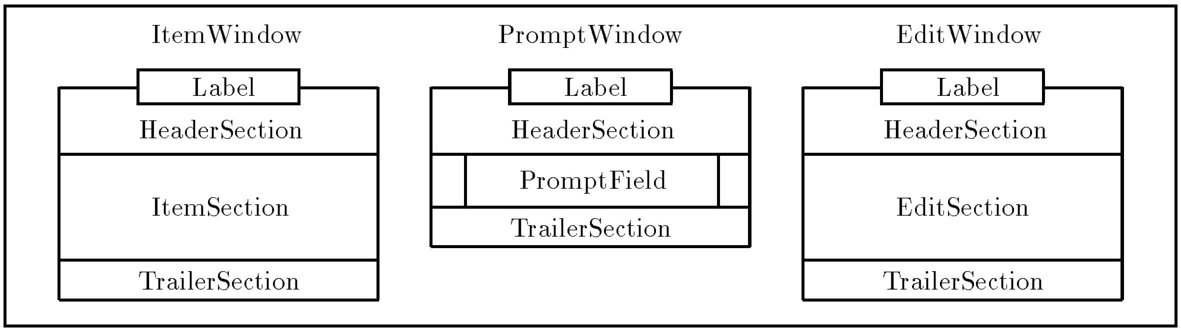

Figure 3.5: Window Sections, `CDFedit`

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.

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

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 only used to display on-line help.

Appendix A

Skeleton Table Format

A.1 Introduction

Skeleton tables are both created by and read by CDF utility programs. `SkeletonTable` creates a skeleton table by reading a CDF. `SkeletonCDF` creates a CDF by reading a skeleton table. In almost all cases the format of the skeleton tables read and written will be the same. Any differences are minor and will be described where appropriate.

The skeleton table has a free format (except where noted) — you need not be concerned with any column alignments, spaces between fields, or spaces between successive lines. However, certain syntax rules do apply to skeleton tables.

1. Lines are limited to 132 characters.
2. Keywords for the header section, gAttributes section, vAttributes section, rVariables section, and end section must always be specified (in that order). The zVariables section is optional — its keyword may be omitted.
3. An exclamation point (`!`) at any point signifies a comment until the end of the line. Any characters encountered after the exclamation point will be ignored. An exclamation point may begin a line (making the entire line a comment). Exclamation points inside delimited character strings are part of the string and do not cause the start of a comment.
4. Attribute and variable names must be delimited. Any character not in the name may be used as the delimiter with the following exceptions:
   (a) Do not use an exclamation point (`!`) to delimit an attribute or variable name.
   (b) Do not use a period (`.`) to delimit an attribute name in the variables section.
   (c) Do not use a left square bracket (`[`) or a numeral to delimit a variable name.
5. When specifying a character string attribute entry value, do not use a hyphen (`-`) to delimit the string or strings (if the string is split across one or more lines).
6. All items are referenced from one (1). These include gAttribute gEntry numbers and NRV variable index values.

In the descriptions that follow, optional fields are shown in brackets ([...]).

A.2 Header Section

The header section contains general information about the CDF. The format of the header section is as follows:

```
#header

                  CDF NAME: <cdf-name>
             DATA ENCODING: <data-encoding>
                  MAJORITY: <variable-majority>
                    FORMAT: <cdf-format>

!     Variables    G.Attributes V.Attributes   Records     Dims       Sizes
!     ---------    ------------ ------------   -------     ----    -----------
   <rVars>/<zVars>   <gAttrs>    <vAttrs>   <n-recs>/z  <n-dims>  <dim-sizes>
```

The fields are defined as follows:

<cdf-name>	The name of the CDF. When SkeletonTable creates a skeleton table, this will be the name of the corresponding CDF (not the full pathname specified). When SkeletonCDF reads a skeleton table, this will be the name of the CDF created unless a CDF pathname is specified on the command line. If the CDF name in the skeleton table is to be used, a full pathname must be specified (if desired) or else the CDF will be created in the default/current directory.
<data-encoding>	The data encoding of the CDF. When specifying a data encoding to the SkeletonCDF program, the following encodings are valid: HOST, NETWORK, VAX, ALPHAVMSd, ALPHAVMSg, SUN, SGi, DECSTATION, ALPHAOSF1, IBMRS, HP, IBMPC, MAC, and NeXT. When a skeleton table is created by SkeletonTable, all of the above encodings with the exception of HOST are possible. Data encoding is described in Section 2.2.8.
<variable-majority>	The variable majority of the CDF. This may be either ROW or COLUMN. Variable majority is described in Section 2.3.12.
<cdf-format>	The format of the CDF. This may be either SINGLE or MULTI. CDF formats are described in Section 2.2.7. Note that this line is optional. Skeleton tables created by SkeletonTable in CDF V2.0 did not have this line because the single-file option did not exist. To allow SkeletonCDF to read skeleton tables created with SkeletonTable in CDF V2.0, this line was made optional. If omitted, SkeletonCDF will create a CDF with the default format for your CDF distribution. Consult your system manager to determine this default. SkeletonTable (in CDF V2.1 and beyond) always generates this line regardless of the version of the CDF being read.

`<rVars>` The number of rVariables in the CDF. `SkeletonTable` always places the correct number here. However, when `SkeletonCDF` reads a skeleton table, this value is ignored (but a place holder is necessary). The number of rVariables created is determined by the number of rVariable definitions in the rVariable definitions section.

`<zVars>` The number of zVariables in the CDF. `SkeletonTable` always places the correct number here. However, when `SkeletonCDF` reads a skeleton table, this value is ignored (but a place holder is necessary). The number of zVariables created is determined by the number of zVariable definitions in the zVariable definitions section.

`<gAttrs>` The number of gAttributes in the CDF. `SkeletonTable` always places the correct number here. However, when `SkeletonCDF` reads a skeleton table, this value is ignored (but a place holder is necessary). The number of gAttributes created is determined by the number of definitions in the gAttributes section.

`<vAttrs>` The number of vAttributes in the CDF. `SkeletonTable` always places the correct number here. However, when `SkeletonCDF` reads a skeleton table, this value is ignored (but a place holder is necessary). The number of vAttributes created is determined by the number of definitions in the vAttributes section.

`<n-recs>` The (maximum) number of rVariable records in the CDF. `SkeletonTable` always places the correct number here. However, when `SkeletonCDF` reads a skeleton table, this value is ignored (but a place holder is necessary). The number of records written to the CDF depends on whether or not any values are specified for variables. NRV variables are described in Section 2.3.10.

`<n-dims>` The number of dimensions for the rVariables in the CDF.

`<dim-sizes>` The dimension sizes for the rVariables in the CDF — one value per dimension. If the rVariables have zero (0) dimensions, this field would be left blank.

An example header section for a CDF with 2-dimensional rVariables follows:

```
#header

                  CDF NAME: sample2
             DATA ENCODING: NETWORK
                  MAJORITY: ROW
                    FORMAT: SINGLE

! Variables G.Attributes V.Attributes Records  Dims   Sizes
! --------- ------------ ------------ -------  ----  -------
    14/0         18            4        1/z      2   180 360
```

If the rVariables had zero dimensions, the header section would be as follows:

```
#header
```

```
                 CDF NAME: sample0
            DATA ENCODING: NETWORK
                 MAJORITY: ROW
                   FORMAT: SINGLE

! Variables G.Attributes V.Attributes Records  Dims  Sizes
! --------- ------------ ------------ -------  ----  -----
    14/0         18             4        1/z      0
```

A.3 gAttributes Section

The gAttributes section contains the definition of each gAttribute as well as any gEntries for those gAttributes. The format of the gAttributes section is as follows:

```
#GLOBALattributes

[<global-scope-attribute-definition>
 <global-scope-attribute-definition>
 <global-scope-attribute-definition>
                  .
                  .
                  .
 <global-scope-attribute-definition>]
```

Where `<global-scope-attribute-definition>`, needless to say, is a gAttribute definition.

Zero or more gAttribute definitions are allowed. (There is no limit on the number of attributes that a CDF may have.) The format of each gAttribute definition is as follows:

```
! Attribute        Entry       Data
! Name             Number      Type           Value
! ---------        ------      ----           -----

  <attr-name>   [<entry-n>:    <data-type>    <value>
                 <entry-n>:   [<data-type>]   <value>
                 <entry-n>:   [<data-type>]   <value>
                     .              .             .
                     .              .             .
                     .              .             .
                 <entry-n>:   [<data-type>]   <value>] .         ! note the "."
```

The fields are defined as follows:

`<attr-name>` The name of the gAttribute. The name must be delimited with a character not appearing in the name itself (e.g., `"TITLE"` or `'History'`). The delimiting characters are not part of the gAttribute name in the CDF.

`<entry-n>` The gEntry number. Zero or more gEntries may be specified for a gAttribute, and there are no restrictions on the gEntry numbers that may be used (except that they must be greater than zero).

`<data-type>` The data type for the gEntry. The data type must be one of the following: CDF_BYTE, CDF_INT1, CDF_UINT1, CDF_INT2, CDF_UINT2, CDF_INT4, CDF_UINT4, CDF_REAL4, CDF_FLOAT, CDF_REAL8, CDF_DOUBLE, CDF_EPOCH, CDF_CHAR, or CDF_UCHAR. The `<data-type>` field is optional for all but the first gEntry specified. If omitted, the data type of the previous gEntry is assumed.

`<value>` The value(s) for the gEntry. A period (`.`) follows the value(s) of the last gEntry for a gAttribute.

Attribute Entry Values

An attribute entry can have more than one element of the specified data type. For character string data types (CDF_CHAR and CDF_UCHAR), each character is an element. The character string must be delimited with a character not appearing in the string itself, and the entire delimited string must be enclosed in braces (e.g., `{ "The CDF title." }`). If the string will not fit on one line, it may be continued on additional lines. The substrings are each delimited with a unique character, and a dash (`-`) is placed at the end (after the terminating delimiter) of each line except the last one. For example,

```
{ "This is a longer " -
  "CDF title that will" -
  " not fit on one line." }
```

For non-character string data types, the elements are enclosed in braces and separated by commas (e.g., `{ 1, 2, 3 }`). If the elements will not all fit on one line, they may be continued on additional lines. For example,

```
{ 1.0, 2.0, 3.0, 4.0, 5.0,
  6.0, 7.0, 8.0, 9.0, 10.0 }
```

Note that an individual element value may not be split across lines.

The format of a value for the CDF_EPOCH data type (which is also considered a non-character string data type) is defined in Section 2.2.10. A CDF_EPOCH value may not be split across two lines.

Several example gAttribute definitions follow:

```
#GLOBALattributes

! Attribute          Entry        Data
! Name               Number       Type         Value
! ---------          ------       ----         -----
```

```
  "TITLEa"              1:    CDF_CHAR     { "CDAW-9A; SABRE" }.

  ^TITLEb^              1:    CDF_CHAR     { "CDAW-9A; SABRE " -
                                             "Backscatter Radar, 20s." }.

  "History"             1:    CDF_CHAR     { "CDF created 02-Jan-1961" }
                        2:                 { "CDF modified 23-Oct-1964" }.

  "TIMES"               1:    CDF_EPOCH    { 04-Jul-1976 12:00:00.000,
                                             31-Oct-1976 00:00:00.000 }
                        2:                 { 25-Dec-1976 01:10:00.000,
                                             01-Jan-1977 01:10:30.000 }.

  &Factors&             1:    CDF_REAL4    { 12.5 }
                        2:                 { 17.4 }
                        3:                 {  8.5 }
                        4:    CDF_INT2     { 7 }
                        5:                 { 12 }.
```

A.4 vAttributes Section

The vAttributes section contains the names of the vAttributes in the CDF. Any rEntries or zEntries for these vAttributes are defined in the rVariables/zVariables sections (following the definition of the corresponding variable). The format of the vAttributes section is as follows:

```
#VARIABLEattributes

[<attribute-name>
 <attribute-name>
 <attribute-name>
        .
        .
        .
 <attribute-name>]
```

Where `<attribute-name>` is a vAttribute name delimited with a character not appearing in the name itself (e.g., `"VALIDMIN"` or `'Units'`). The delimiting characters are not part of the vAttribute name in the CDF. There may be zero or more vAttribute names. (There is no limit on the number of attributes that a CDF may have.)

An example vAttributes section follows:

```
#VARIABLEattributes

  "FIELDNAM"
  "VALIDMIN"
  "Units"
```

A.5 rVariables Section

The rVariables section contains the definition of each rVariable in the CDF, the values for any vAttribute rEntries associated with each rVariable, and (optionally) data values for those rVariables. The format of the rVariables section is as follows:

```
#variables

[<variable-definition>
 <variable-definition>
 <variable-definition>
           .
           .
           .
 <variable-definition>]
```

Where `<variable-definition>` is an rVariable definition. The format of each rVariable definition is as follows:

```
! Variable            Data         Number     Record     Dimension
! Name                Type        Elements   Variance    Variances
! --------            ----        --------   --------    ---------

  <var-name>     <var-data-type>  <n-elems> <rec-vary>  <dim-varys>

  !  Attribute           Data
  !  Name                Type           Value
  !  ---------           ----           -----

     [<attr-name>  <entry-data-type>    <entry-value>
      <attr-name>  <entry-data-type>    <entry-value>
      <attr-name>  <entry-data-type>    <entry-value>
           .               .                  .
           .               .                  .
           .               .                  .
      <attr-name>  <entry-data-type>    <entry-value>].        ! Note the "."

   [[<rec-num>:]<indices> = <value>
    [<rec-num>:]<indices> = <value>
    [<rec-num>:]<indices> = <value>
        .            .
        .            .
        .            .
    [<rec-num>:]<indices> = <value>]
```

Each field is defined as follows:

`<var-name>` The name of the rVariable. The name must be delimited with a character

	not appearing in the name itself (e.g., "EPOCH" or 'Temperature'). The delimiting characters are not part of the rVariable name in the CDF.
`<var-data-type>`	The data type for the rVariable. The data type must be one of the following: CDF_BYTE, CDF_INT1, CDF_UINT1, CDF_INT2, CDF_UINT2, CDF_INT4, CDF_UINT4, CDF_REAL4, CDF_FLOAT, CDF_REAL8, CDF_DOUBLE, CDF_EPOCH, CDF_CHAR, or CDF_UCHAR.
`<n-elems>`	The number of elements of the data type. For character string data types (CDF_CHAR and CDF_UCHAR), this is the number of characters in each string. For non-character string data types, this value must be one (1).
`<rec-vary>`	The record variance of the rVariable. This must be either T (the values vary from record to record) or F (the values do not vary from record to record).
`<dim-varys>`	The dimension variances of the rVariable. For each dimension there must be either a T (the values vary along that dimension) or F (the values do not vary along that dimension). Each dimension variance must be separated by at least one space. If the rVariables have zero dimensions, this field would be left blank.
`<attr-name>`	The name of the vAttribute for which to specify an rEntry for this rVariable. The vAttribute must have been specified in the vAttributes section. The name must be delimited with a character not appearing in the name itself (e.g., "SCALEMAX" or 'range'). The delimiting characters are not part of the vAttribute name in the CDF.
`<entry-data-type>`	The data type for the vAttribute rEntry. The data type must be one of the following: CDF_BYTE, CDF_INT1, CDF_UINT1, CDF_INT2, CDF_UINT2, CDF_INT4, CDF_UINT4, CDF_REAL4, CDF_FLOAT, CDF_REAL8, CDF_DOUBLE, CDF_EPOCH, CDF_CHAR, or CDF_UCHAR.
`<entry-value>`	The value(s) for the vAttribute rEntry. The format of attribute entry values is described in Section A.3. **NOTE:** The last rEntry MUST be followed by a period (.). If no rEntries are specified for an rVariable, the period must still be present.
`<rec-num>`	The record number of an rVariable value. This will be present only for record-variant (RV) rVariables.
`<indices>`	The indices of an rVariable value. The indices are enclosed in brackets and separated by commas (e.g., [23,1] or [1,80]). If the rVariables have zero dimensions, [] would be specified (the brackets are still required).
`<value>`	The value at the given record/indices. For character string data types (CDF_CHAR or CDF_UCHAR) the string must be delimited with a unique character and enclosed in braces ({...}) in the same manner as for an attribute entry for a character string data type. For non-character string data types the value is not enclosed in braces (the braces are not necessary because there can only be one element). The format for CDF_EPOCH values is described in Section 2.2.10.

The vAttribute rEntries are optional. If omitted, the terminating period is still required. The rVariable values are also optional.

Several sample rVariable definitions for a CDF with 2-dimensional rVariables follow:

```
! Variable           Data       Number     Record    Dimension
! Name               Type      Elements   Variance   Variances
! --------           ----      --------   --------   ---------

  "Latitude"       CDF_REAL4       1          F          F T

  ! Attribute        Data
  ! Name             Type        Value
  ! --------         ----        -----

    "VALIDMIN"     CDF_REAL4    { -90.0 }
    "VALIDMAX"     CDF_REAL4    { 90.0 }
    "scale"        CDF_REAL4    { -60.0, 60.0 }.

    [1,1] = -60.0
    [1,2] = -30.0
    [1,3] = 0.0
    [1,4] = 30.0
    [1,5] = 60.0


! Variable           Data       Number     Record    Dimension
! Name               Type      Elements   Variance   Variances
! --------           ----      --------   --------   ---------

  "EPOCH"          CDF_EPOCH       1          T          F F

  ! Attribute        Data
  ! Name             Type        Value
  ! --------         ----        -----

    "scale"        CDF_EPOCH    { 10-Oct-1991 00:00:00.000,
                                  20-Oct-1991 23:59:59.999 }.


! Variable           Data       Number     Record    Dimension
! Name               Type      Elements   Variance   Variances
! --------           ----      --------   --------   ---------

  'Tmp'            CDF_INT2        1          T          T T

  ! Attribute        Data
  ! Name             Type        Value
  ! --------         ----        -----

    'Fieldname'    CDF_CHAR     { "Temperature (C)" }.
```

```
! Variable           Data       Number     Record     Dimension
! Name               Type       Elements   Variance   Variances
! --------           ----       --------   --------   ---------

  "Pres_lvl"       CDF_REAL4       1          T          F F

  ! Attribute        Data
  ! Name             Type       Value
  ! --------         ----       -----

    .   ! no attribute entries

    1:[1,1] = 1013.1
    2:[1,1] = 1015.0
    3:[1,1] = 1012.3
```

A sample variable definition for a CDF with 0-dimensional rVariables follows:

```
! Variable           Data       Number     Record     Dimension
! Name               Type       Elements   Variance   Variances
! --------           ----       --------   --------   ---------

  "Latitude"       CDF_REAL4       1          F

  ! Attribute        Data
  ! Name             Type       Value
  ! --------         ----       -----

    "VALIDMIN"     CDF_REAL4    { -90.0 }
    "VALIDMAX"     CDF_REAL4    { 90.0 }.

    [] = -12.3
```

A.6 zVariables Section

The optional zVariables section contains the definition of each zVariable in the CDF, the values for any vAttribute zEntries associated with each zVariable, and (optionally) data values for those zVariables. The format of the zVariables section is as follows:

```
#zVariables

[<variable-definition>
 <variable-definition>
 <variable-definition>
           .
```

```
         .
         .
 <variable-definition>]
```

Where `<variable-definition>` is a zVariable definition. The format of each zVariable definition is as follows:

```
! Variable        Data       Number                     Record    Dimension
! Name            Type      Elements   Dims   Sizes   Variance   Variances
! --------        ----      --------   ----   -----   --------   ---------

  <var-name> <var-data-type> <n-elems> <dims> <sizes> <rec-vary> <dim-varys>

  !  Attribute           Data
  !  Name                Type           Value
  !  ---------           ----           -----

     [<attr-name>  <entry-data-type>    <entry-value>
      <attr-name>  <entry-data-type>    <entry-value>
      <attr-name>  <entry-data-type>    <entry-value>
           .               .                   .
           .               .                   .
           .               .                   .
      <attr-name>  <entry-data-type>    <entry-value>].        ! Note the "."

   [[<rec-num>:]<indices> = <value>
    [<rec-num>:]<indices> = <value>
    [<rec-num>:]<indices> = <value>
        .            .
        .            .
        .            .
    [<rec-num>:]<indices> = <value>]
```

Each field is defined as follows:

`<var-name>` The name of the zVariable. The name must be delimited with a character not appearing in the name itself (e.g., `"EPOCH"` or `'Temperature'`). The delimiting characters are not part of the zVariable name in the CDF.

`<var-data-type>` The data type for the zVariable. The data type must be one of the following: `CDF_BYTE`, `CDF_INT1`, `CDF_UINT1`, `CDF_INT2`, `CDF_UINT2`, `CDF_INT4`, `CDF_UINT4`, `CDF_REAL4`, `CDF_FLOAT`, `CDF_REAL8`, `CDF_DOUBLE`, `CDF_EPOCH`, `CDF_CHAR`, or `CDF_UCHAR`.

`<n-elems>` The number of elements of the data type. For character string data types (`CDF_CHAR` and `CDF_UCHAR`) this is the number of characters in each string. For non-character string data types this value must be one (1).

`<dims>` The number of dimensions for the zVariable.

<sizes> — The dimension sizes — one value per dimension. If the zVariable has zero (0) dimensions, this field would be left blank.

<rec-vary> — The record variance of the zVariable. This must be either T (the values vary from record to record) or F (the values do not vary from record to record).

<dim-varys> — The dimension variances of the zVariable. For each dimension there must be either a T (the values vary along that dimension) or F (the values do not vary along that dimension). Each dimension variance must be separated by at least one space. If the zVariable has zero dimensions, this field would be left blank.

<attr-name> — The name of the vAttribute for which to specify a zEntry for this zVariable. The vAttribute must have been specified in the vAttributes section. The name must be delimited with a character not appearing in the name itself (e.g., "SCALEMAX" or 'range'). The delimiting characters are not part of the vAttribute name in the CDF.

<entry-data-type> — The data type for the vAttribute zEntry. The data type must be one of the following: CDF_BYTE, CDF_INT1, CDF_UINT1, CDF_INT2, CDF_UINT2, CDF_INT4, CDF_UINT4, CDF_REAL4, CDF_FLOAT, CDF_REAL8, CDF_DOUBLE, CDF_EPOCH, CDF_CHAR, or CDF_UCHAR.

<entry-value> — The value(s) for the vAttribute zEntry. The format of attribute entry values is described in Section A.3.

NOTE: The last zEntry MUST be followed by a period (.). If no zEntries are specified for a zVariable, the period must still be present.

<rec-num> — The record number of an zVariable value. This will be present only for record-variant (RV) zVariables.

<indices> — The indices of an zVariable value. The indices are enclosed in brackets and separated by commas (e.g., [23,1] or [1,80]). If the zVariable has zero dimensions, [] would be specified (the brackets are still required).

<value> — The value at the given record/indices. For character string data types (CDF_CHAR or CDF_UCHAR) the string must be delimited with a unique character and enclosed in braces ({...}) in the same manner as for an attribute entry for a character string data type. For non-character string data types the value is not enclosed in braces (the braces are not necessary because there can only be one element). The format for CDF_EPOCH values is described in Section 2.2.10.

The vAttribute zEntries are optional. If omitted, the terminating period is still required. The zVariables values are also optional.

Several sample zVariable definitions follow:

```
! Variable           Data      Number                    Record   Dimension
! Name               Type     Elements  Dims  Sizes  Variance  Variances
! --------           ----     --------  ----  -----  --------  ---------
```

```
  "Instrument"    CDF_CHAR      10       0              F

  ! Attribute       Data
  ! Name            Type       Value
  ! --------        ----       -----

    "FIELDNAM"    CDF_CHAR     { "Measuring instrument" }.

  [] = { "Gonkulator" }


! Variable          Data      Number                 Record   Dimension
! Name              Type     Elements  Dims  Sizes  Variance  Variances
! --------          ----     --------  ----  -----  --------  ---------

  "Ticks"         CDF_BYTE       1       1     3        T         T

  ! Attribute       Data
  ! Name            Type       Value
  ! --------        ----       -----

  .    ! no attribute entries

  1:[1] = 1
  1:[2] = 2
  1:[3] = 3
  2:[1] = 3
  2:[2] = 2
  2:[3] = 1


! Variable           Data      Number                      Record   Dimension
! Name               Type     Elements  Dims    Sizes     Variance  Variances
! --------           ----     --------  ----  ----------  --------  ---------

  "WIND VELOCITY"  CDF_REAL4      1       3   360 180 10      T       T T T

  ! Attribute       Data
  ! Name            Type       Value
  ! --------        ----       -----

    "FIELDNAM"    CDF_CHAR     { "Wind velocity." }
    "VALIDMIN"    CDF_REAL4    { 0.0 }
    "VALIDMAX"    CDF_REAL4    { 300.0 }
    "UNITS   "    CDF_CHAR     { "Knots" }
    "FORMAT  "    CDF_CHAR     { "F9.1    " }.
```

A.7 End Section

This section simply consists of the keyword `#end`. This section is required.

A.8 Example Skeleton Table

An example skeleton table containing rVariables and zVariables follows:

```
! Skeleton table for the "example2" CDF.
! Generated: Thursday, 17-Nov-1994 14:07:58
! CDF created/modified by CDF V2.4.10
! Skeleton table created by CDF V2.5.0


#header

                  CDF NAME: example2
             DATA ENCODING: NETWORK
                  MAJORITY: ROW
                    FORMAT: SINGLE

! Variables G.Attributes V.Attributes Records  Dims  Sizes
! --------- ------------ ------------ -------  ----  -----
      4          1             7         1       2   11 7


#GLOBALattributes

! Attribute         Entry       Data
! Name              Number      Type       Value
! ---------         ------      ----       -----

  "TITLE"               1:    CDF_CHAR     { "Title for example2 CDF." } .


#VARIABLEattributes

  "FIELDNAM"
  "VALIDMIN"
  "VALIDMAX"
  "SCALEMIN"
  "SCALEMAX"
  "UNITS"
  "FORMAT"


#variables
```

```
! Variable           Data       Number     Record     Dimension
! Name               Type      Elements   Variance   Variances
! --------           ----      --------   --------   ---------

  "EPOCH"          CDF_EPOCH      1          T          F F

  ! Attribute        Data
  ! Name             Type       Value
  ! --------         ----       -----

    "FIELDNAM"     CDF_CHAR     { "Time since 0 A.D.   " }
    "VALIDMIN"     CDF_EPOCH    { 01-Jan-0000 00:00:00.000 }
    "VALIDMAX"     CDF_EPOCH    { 01-Jan-2089 00:00:00.000 }
    "SCALEMIN"     CDF_EPOCH    { 01-Apr-1986 07:00:00.000 }
    "SCALEMAX"     CDF_EPOCH    { 01-Apr-1986 23:00:00.000 }
    "UNITS"        CDF_CHAR     { "milliseconds (UT)   " }
    "FORMAT"       CDF_CHAR     { "E14.0   " } .

! Variable           Data       Number     Record     Dimension
! Name               Type      Elements   Variance   Variances
! --------           ----      --------   --------   ---------

  "LONGITUD"       CDF_REAL4      1          F          T F

  ! Attribute        Data
  ! Name             Type       Value
  ! --------         ----       -----

    "FIELDNAM"     CDF_CHAR     { "Longitude variable  " }
    "VALIDMIN"     CDF_REAL4    { 0.0 }
    "VALIDMAX"     CDF_REAL4    { 180.0 }
    "SCALEMIN"     CDF_REAL4    { -50.0 }
    "SCALEMAX"     CDF_REAL4    { 50.0 }
    "UNITS"        CDF_CHAR     { "Degrees             " }
    "FORMAT"       CDF_CHAR     { "F8.3    " } .

    [1,1] = -50.0
    [2,1] = -40.0
    [3,1] = -30.0
    [4,1] = -20.0
    [5,1] = -10.0
    [6,1] = 0.0
    [7,1] = 10.0
    [8,1] = 20.0
    [9,1] = 30.0
    [10,1] = 40.0
    [11,1] = 50.0

! Variable           Data       Number     Record     Dimension
! Name               Type      Elements   Variance   Variances
```

```
! --------            ----      --------   --------   ---------

  "LATITUDE"        CDF_REAL4       1          F         F T

  ! Attribute         Data
  ! Name              Type       Value
  ! --------          ----       -----

    "FIELDNAM"      CDF_CHAR     { "Latitude variable   " }
    "VALIDMIN"      CDF_REAL4    { 0.0 }
    "VALIDMAX"      CDF_REAL4    { 90.0 }
    "SCALEMIN"      CDF_REAL4    { -30.0 }
    "SCALEMAX"      CDF_REAL4    { 30.0 }
    "UNITS"         CDF_CHAR     { "Degrees             " }
    "FORMAT"        CDF_CHAR     { "F8.3    " } .

    [1,1] = -30.0
    [1,2] = -20.0
    [1,3] = -10.0
    [1,4] = 0.0
    [1,5] = 10.0
    [1,6] = 20.0
    [1,7] = 30.0

! Variable            Data      Number     Record     Dimension
! Name                Type      Elements   Variance   Variances
! --------            ----      --------   --------   ---------

  "TEMPERATURE"     CDF_INT4        1          T         T T

  ! Attribute         Data
  ! Name              Type       Value
  ! --------          ----       -----

    "FIELDNAM"      CDF_CHAR     { "Temperature         " }
    "VALIDMIN"      CDF_INT4     { 0 }
    "VALIDMAX"      CDF_INT4     { 50 }
    "SCALEMIN"      CDF_INT4     { 0 }
    "SCALEMAX"      CDF_INT4     { 10 }
    "UNITS"         CDF_CHAR     { "Deg C               " }
    "FORMAT"        CDF_CHAR     { "I2      " } .


#zVariables

! Variable            Data     Number                 Record   Dimension
! Name                Type     Elements  Dims  Sizes  Variance  Variances
! --------            ----     --------  ----  -----  --------  ---------

  "BIAS"            CDF_INT4       1       0              T
```

```
  ! Attribute       Data
  ! Name            Type       Value
  ! --------        ----       -----

    "FIELDNAM"    CDF_CHAR     { "Correction bias for temperature" }
    "VALIDMIN"    CDF_INT4     { -5 }
    "VALIDMAX"    CDF_INT4     { 5 }
    "UNITS   "    CDF_CHAR     { "deg C               " }
    "FORMAT  "    CDF_CHAR     { "I2      " } .

    1:[] = 34
    2:[] = 28
    3:[] = 17


! Variable          Data      Number                 Record   Dimension
! Name              Type     Elements  Dims  Sizes  Variance  Variances
! --------          ----     --------  ----  -----  --------  ---------

  "Coefficients"  CDF_REAL4      1       1     3        F         T

  ! Attribute       Data
  ! Name            Type       Value
  ! --------        ----       -----

    "FIELDNAM"    CDF_CHAR     { "Temperature model coefficients." }
    "FORMAT  "    CDF_CHAR     { "F9.1    " } .

  [1] = -0.0254
  [2] = 14.2338
  [3] = -9.9444


! Variable          Data      Number                 Record   Dimension
! Name              Type     Elements  Dims  Sizes  Variance  Variances
! --------          ----     --------  ---- ------- --------  ---------

  "TMP-model"     CDF_REAL4      1       2  360 180     T        T T

  ! Attribute       Data
  ! Name            Type       Value
  ! --------        ----       -----

    "FIELDNAM"    CDF_CHAR     { "Temperature model." }
    "VALIDMIN"    CDF_REAL4    { -20.0 }
    "VALIDMAX"    CDF_REAL4    { 50.0 }
    "SCALEMIN"    CDF_REAL4    { 0.0 }
    "SCALEMAX"    CDF_REAL4    { 30.0 }
    "UNITS   "    CDF_CHAR     { "deg C               " }
    "FORMAT  "    CDF_CHAR     { "F9.6    " } .
```

```
#end
```

Appendix B

IDL Support

B.1 Introduction

The CDF distribution includes a set of IDL functions/procedures that allow access to the CDF library (hereafter referred to as the CDF/IDL interface). The functions/procedures are used in the same manner as the Standard and Internal Interfaces are used in programs written in C. CDFs can be both read and written/modified.

NOTE: There is, however, one difference when using the Internal Interface (`CDFlib`). The current CDF must be selected each time `CDFlib` is called. This is because of how IDL does the dynamic mapping to the CDF library at run-time. Failure to select the current CDF may result in IDL crashing.

Research Systems, Inc. (the developers of IDL) have also implemented an interface to CDF that is part of the IDL product. It differs from the interface provided with the CDF distribution in that it is intended more for the non-programmer (and is functionally similar to other interfaces they provide). Which interface you choose to use will depend on your personal preferences.

CDF's interface relies on IDL's support of dynamic linking. IDL in turn relies on the operating system being used to support dynamic linking in a reasonable way. Currently, only the VAX (VMS), DEC Alpha (OpenVMS & OSF/1), Sun (SunOS & SOLARIS), HP (HP-UX), and IBM RS6000 (AIX) support dynamic linking in a way that makes it possible to use CDF's interface to IDL. Because IDL's CDF interface is built into the IDL executable, it does not rely on dynamic linking and is therefore available on every machine on which IDL is supported.

B.2 Using CDF's IDL Interface

This section assumes that you are using the "definitions" file provided with the CDF distribution (and that it has been properly configured and named by your system manager). The "definitions" file is a command file named `DEFINITIONS.COM` on VMS systems and a script file named `definitions.<shell>` on UNIX systems (where `<shell>` is the shell being used). The "definitions" file defines the logical name `IDL_STARTUP` on VMS systems and the environment variable `IDL_STARTUP` on UNIX systems to the full directory path of an