Acucorp's Indexed File Record Editor

alfred

Acucorp, Inc.
8515 Miralani Drive
San Diego, CA 92126
858.689.4500

E-01-RM-070222-Alf-8.0

Contents

1 Acucorp’s Indexed File Record Editor — alfred

Key Topics

1.1 Introduction

As of Version 8.0, Acucorp's Indexed File Record Editor (alfred) is provided as a sample program and is located in the "sample" folder under "AcuGT".

alfred is an interactive, menu-driven, record-level editor for use with indexed files. It can be used to edit any indexed file supported by your ACUCOBOL-GT runtime.

alfred allows you to view, modify, add, or delete individual records stored in an indexed file. It's especially valuable to developers engaged in testing and debugging data file layout and file I/O actions.

alfred is also valuable to application administrators who need to make occasional, record-level changes to data files. It is *not* well suited to making large scale, global or repetitive changes.

alfred incorporates floating windows, dialog boxes, and bitmapped buttons in its user interface. It uses character-mode emulation to present the graphical interface in character-mode environments.

1.2 Modes of Operation

With XFD file

If you've built a data dictionary (also known as an Extended File Descriptor or XFD) for your indexed file, **alfred** can access the data dictionary to obtain specific information about the fields in each record, and can display the records according to their field structure. This makes the file easy to read on your screen and all editing is done at the field level.

To build an XFD file, specify the "-Fx" option when you compile your COBOL program. For more information about XFD files, their creation and use, see the *AcuCOBOL-GT User's Guide*, section 5.3, "XFD Files."

Although ACUCOBOL-GT allows Vision files to have as many as 119 alternate keys, memory limitations restrict **alfred** to XFD files having no more than 114 alternate keys. Attempting to open an XFD file containing more than 114 alternate keys results in a memory access error.

Without XFD file

If you do not have an XFD file, **alfred** can still access and display the records. However, it has no field-level information about the file and, therefore, data is presented in groups of characters. All user interactions are done at the character level, either in ASCII, hexadecimal, or octal.

1.3 Security

Because this utility can significantly alter data files, we recommend it be made available to qualified developers and data file administrators only. If possible, restrict access to **alfred** with operating system file access protections. UNIX sites can take advantage of an additional layer of security provided by Acucorp, as described below.

1.3.1 UNIX security

On UNIX systems, Acucorp provides an extra security mechanism for **alfred** users. Your site may choose to set up either a list of users who are allowed to use **alfred**, or a list of users who are denied access to **alfred**. This security mechanism is described in the following paragraphs.

If the text file “/etc/alfred.allow” exists, then a user may access **alfred** only if the user’s login name is listed in the file. Thus, an empty “/etc/alfred.allow” file denies access to everyone. **alfred** does not search for a “deny” file if the file “/etc/alfred.allow” exists.

If the “allow” file does *not* exist, then **alfred** checks for the existence of a “deny” file. This file must be named “/etc/alfred.deny”. Users whose names appear in the “deny” file are *not* allowed to use **alfred**. Note that, if no “allow” file exists, an empty “deny” file gives access to everyone.

If neither an "allow" file nor a "deny" file exists, then only root can use **alfred**.

1.4 Translation of Messages

The error messages and help messages that **alfred** displays are grouped into a single file. This makes it easier to translate **alfred** error messages into other languages. The runtime configuration variable ALFRED-MESSAGE-FILE holds the full path and name of the message file. Set this variable to the path and name of the file you want to use for messages. For example:

```
ALFRED-MESSAGE-FILE   /etc/alfred.msg
```

or

```
ALFRED-MESSAGE-FILE   c:\acucobol\etc\alfred.msg
```

The default value of this variable is "alfred.msg".

1.5 Starting alfred

When **alfred** is started, it needs to know a number of things about the file to be edited before it can display file records or support record editing.

1. **alfred** needs to know the name of the data file to be edited.
2. If it's available, **alfred** will use the data file's companion data dictionary (XFD file) to identify the fields in each record. This allows **alfred** to display each record in a format that is more meaningful and easier to read. **alfred** automatically looks for a data dictionary when the data file is specified, but you can also specify the XFD file explicitly at startup.
3. If a data dictionary does not exist, **alfred** prompts for *keymap* information. The keymap information must describe the type of data at each character position in the key.
4. **alfred** needs to know which key to use to sort the records.

The following sections describe how these four items are entered.

1.5.1 Initial startup and filename designation

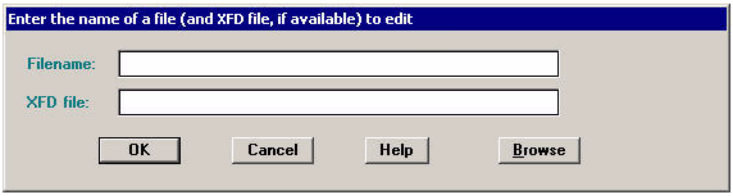
To start **alfred** under UNIX or Linux, at the operating system command line enter:

```
runcbl alfred [datafile [XFDfile]]
```

To start **alfred** in a graphical environment, double click on the **alfred** icon, or use the command line option, such as "Run" in the Program Manager File menu or the Start menu under Microsoft Windows, and enter:

```
wrun32.exe alfred [datafile [XFDfile]]
```

datafile is the name of the indexed file to be edited. *XFDfile* is the name of the XFD file associated with datafile. Both datafile and XFDfile are optional. However, if you start **alfred** without specifying a data file name, you will be prompted for a filename and an XFD at the beginning of the program:



Enter the name of a file (and XFD file, if available) to edit

alfred does *not* use the FILE_PREFIX and FILE_SUFFIX configuration variables. Therefore you need to specify either a full path, or a relative path from the current directory, when you specify file names. You must also include the filename suffix. Note that by default, data files are opened in *read-only* mode.

On hosts that support a file system browsing function, a Browse push button is included in the bottom right of the File Open dialog box (as shown above). Click the Browse button to initiate the browse activity.

After you enter a filename, even if you do not specify an XFD file, **alfred** will try to locate a matching XFD file. In order for **alfred** to begin its search, you must move to the XFD File field on the screen. You may tab over to the field, or click on it with the mouse. Then select "OK" when you are ready for the search to begin. It looks for an XFD file that has the same base name as the data file, plus the extension ".xfd". It first looks for the XFD in the directory specified for the data file. It then looks in the current directory. Lastly, it looks in the directory specified in the runtime configuration variable XFD-DIRECTORY. If **alfred** finds a matching XFD file, the next prompt you see is for the key selection. If **alfred** does not find an XFD file, you are prompted for key information.

Note: **alfred** checks to make sure that the XFD and the data file match in record size, number of keys, size of keys, and so forth. If they do not match, **alfred** will display a warning message, and will not use the XFD file.

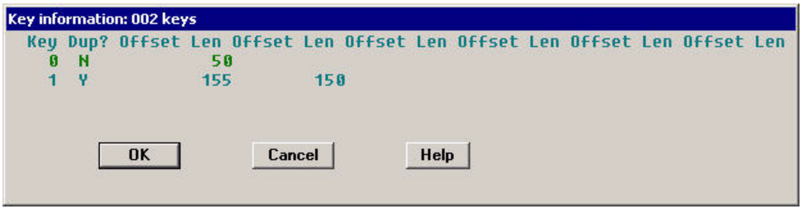
Supplying an XFD file to use with the datafile gives **alfred** valuable information about the fields, which makes the screens of data more readable, and makes editing much easier. See the *AcuCOBOL-GT User's Guide*, section 5.3, "XFD Files," for more information about XFD files.

1.5.2 Specifying the key

Whether you designate an XFD file or not, **alfred** needs to know which key to use to sort the records.

alfred queries the file about its key structure, and displays this information to the user, one key per line.

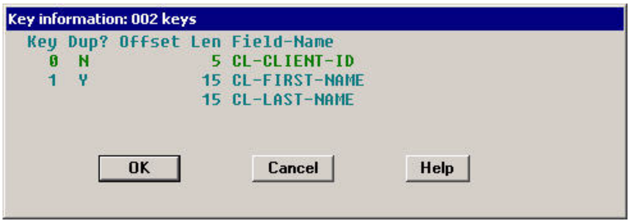
If no XFD is available, the display looks like this:



Key information: 002 keys

```
 Key Dup? Offset Len Offset Len Offset Len Offset Len Offset Len Offset Len
   0  N           50
   1  Y          155        150
```

If an XFD file has been specified, the key display for the same file looks like this:



Key information: 002 keys

```
Key Dup? Offset Len Field-Name
  0  N            5 CL-CLIENT-ID
  1  Y           15 CL-FIRST-NAME
                 15 CL-LAST-NAME
```

On a graphical system, select a key by clicking on an item in the Key list. Then click **OK** or press **Enter** to accept the key. On a character-based display, use the mouse or the arrow keys to select the desired key, and press **Enter** to accept the key. The selected key is used to sort the file.

On systems without mouse support, you can navigate the screen and select a key with the following special keys:

1. The Up arrow highlights the previous key.
2. The Down arrow highlights the next key.
3. The Page Up displays the previous page of keys.
4. The Page Down displays the next page of keys.
5. The Home key returns to the first key.
6. The End key positions the cursor at the last key.

When you have selected the key, if you've designated an XFD file, **alfred** has all the information it needs to display the data. You are now ready to edit the file. For a discussion of editing options, proceed to section 1.6 "Editing Records," below. If you are not using an XFD file, you'll need to provide certain *keymap* information, as discussed in the next section.

1.5.3 The keymap

If you've chosen a key, but are not using an XFD file, **alfred** needs information about how to accept and display the key. This information is referred to as the *keymap* for the key. A keymap is a series of As, Ds, Os, and Xs, one character for each byte in the key.

Each character in the keymap describes the type of data **alfred** should find at the corresponding character position.

A An "A" indicates alphanumeric data.

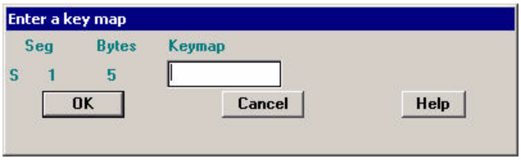
D A "D" indicates binary data that should be accepted and displayed in decimal format.

O An "O" indicates binary data that should be accepted and displayed in octal format.

X An "X" indicates binary data that should be accepted and displayed in hexadecimal format.

A series of characters can be represented with *count notation*, like COBOL picture clauses. For example, a keymap of "AAAAA" is equivalent to "A(5)".

alfred prompts for keymap information by displaying the following input screen:


Enter a key map

	Seg	Bytes	Keymap
S	1	5	

You need to enter a keymap for each segment of the key. Each keymap must have non-space characters equal to the number of bytes indicated under "Bytes".

Here's an example:

Suppose the first segment of the key is a PIC 9(5) field, and the second segment is composed of two fields, a PIC X(15) and a PIC 9(8)V99 COMP-6.

You could enter the following keymap:

A(5)	for the first segment, and
A(15) X(5)	for the second segment

Sometimes segments of a key are composed of different types of data, even though they are known to the file to be contiguous. For clarity, spaces are therefore allowed in keymap descriptions; the spaces themselves do not have value, but are just indicators that the actual segment is to be viewed as multiple logical segments.

If the keymap entered is not valid, a character is displayed in column one of the segment line. The character “S” indicates that the keymap is incomplete. The character “C” indicates that the keymap contains an invalid character.

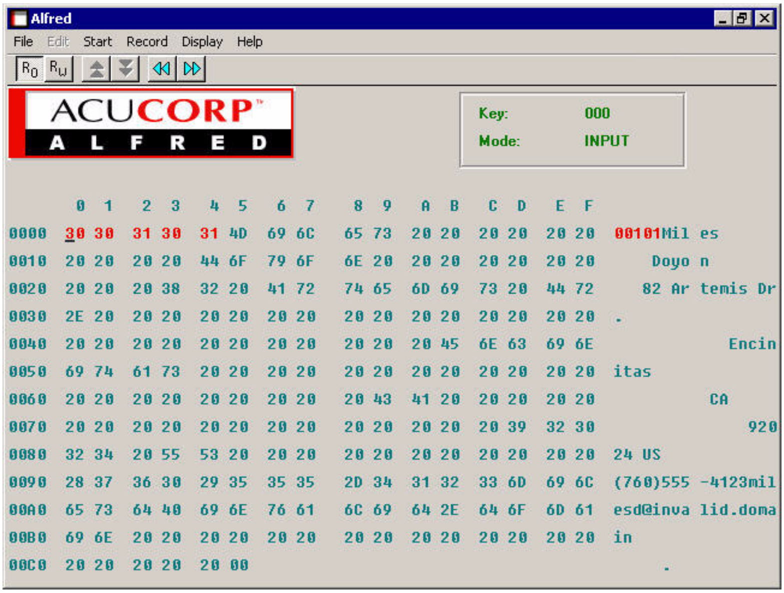
It is possible to enter more characters for a keymap than the segment contains, so all keymap entries are checked for size as well as type.

1.6 Editing Records

When **alfred** has the information it needs, it displays the first record. *By default, data files are opened in read-only mode.* You cannot alter a record in the file until you open it read-write. To do this, access the File menu with the mouse or the Esc key and choose **Read/Write**. On graphical systems, you can also click the **RW** button on the toolbar.

All menu bar options are described in section 3.5.5.3. Before we discuss these options, let’s take a look at the way that **alfred** displays data. The display differs significantly, depending on whether an XFD is used.

1.6.1 Editing without an XFD



Key: 000
Mode: INPUT

```
      0  1   2  3   4  5   6  7    8  9   A  B   C  D   E  F
0000  30 30  31 30  31 4D  69 6C   65 73  20 20  20 20  20 20  00101Mil es
0010  20 20  20 20  44 6F  79 6F   6E 20  20 20  20 20  20 20      Doyo n
0020  20 20  20 38  32 20  41 72   74 65  6D 69  73 20  44 72     82 Ar temis Dr
0030  2E 20  20 20  20 20  20 20   20 20  20 20  20 20  20 20  .
0040  20 20  20 20  20 20  20 20   20 20  20 45  6E 63  69 6E              Encin
0050  69 74  61 73  20 20  20 20   20 20  20 20  20 20  20 20  itas
0060  20 20  20 20  20 20  20 20   20 43  41 20  20 20  20 20            CA
0070  20 20  20 20  20 20  20 20   20 20  20 20  20 39  32 30                920
0080  32 34  20 55  53 20  20 20   20 20  20 20  20 20  20 20  24 US
0090  28 37  36 30  29 35  35 35   2D 34  31 32  33 6D  69 6C  (760)555 -4123mil
00A0  65 73  64 40  69 6E  76 61   6C 69  64 2E  64 6F  6D 61  esd@inva lid.doma
00B0  69 6E  20 20  20 20  20 20   20 20  20 20  20 20  20 20  in
00C0  20 20  20 20  20 00                                           .
```

If there's no XFD file, the record is displayed in two formats. On the left side of the screen, the record is displayed in hexadecimal or octal (hexadecimal is the default). On the right side of the screen, the printable characters are displayed as themselves, and non-printable characters are displayed as a period (".").

You can use the mouse or special keys to move around the screen and change the display. The special keys work as follows:

1. The Arrow keys move around the screen.
2. The Page Up/Page Down keys display the previous/next page.
3. The Home key goes to the first byte of the record.
4. The End key goes to the last byte of the record.
5. Ctrl-X undoes the last change. (Repeatedly pressing Ctrl-X will continue to undo the last change, up to 21,000 single-byte modifications.)

6. Ctrl-T toggles between Hex and Octal display.

7. Ctrl-N gets the next record.

8. Ctrl-P gets the previous record.

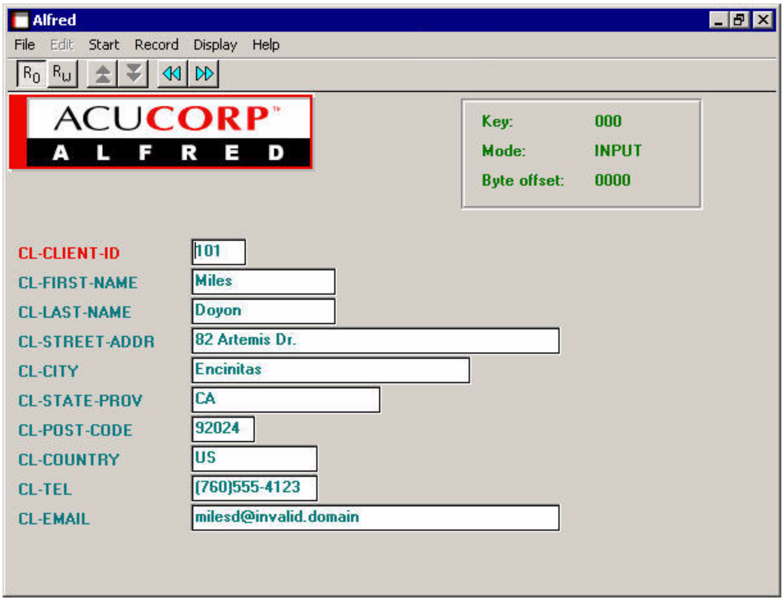
9. The Tab key toggles between the binary data and the ASCII data.

In the ASCII display, you can enter any printable character. In the binary display, you can enter hexadecimal or octal characters only.

1.6.2 Editing with an XFD

This section describes how editing is performed when **alfred** has field-level information about the file (an XFD). In this case, all editing is done at the field level.

The data display typically looks like the following screen illustration.



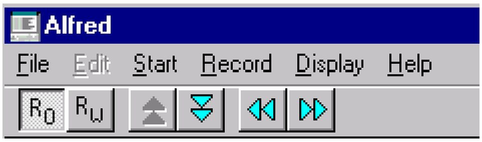
Key:	000
Mode:	INPUT
Byte offset:	0000

CL-CLIENT-ID	101
CL-FIRST-NAME	Miles
CL-LAST-NAME	Doyon
CL-STREET-ADDR	82 Artemis Dr.
CL-CITY	Encinitas
CL-STATE-PROV	CA
CL-POST-CODE	92024
CL-COUNTRY	US
CL-TEL	(760)555-4123
CL-EMAIL	milesd@invalid.domain

The option Evaluate Conditions becomes available under the Display menu. If you change a field that is the controlling value for a WHEN directive in the XFD, the display of data associated with that directive will *not* be updated to match the new condition until you select Evaluate Conditions.

1.6.3 Menu bar choices

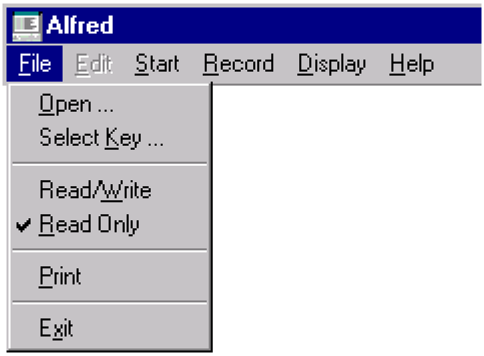
The following menu bar and, on graphical systems, toolbar are always available:



On character-based systems, use the Esc key, or mouse, if supported, to activate the menu bar.

Depending on the current state of the editor, some items on the main menu or submenus may be displayed at half intensity, or gray, or in parentheses. Such menu items are unavailable, because they are invalid given the program state.

1.6.4 The File menu



Open prompts for the name of a new file to edit.

Select Key displays a dialog box that allows you to select a new sort key.

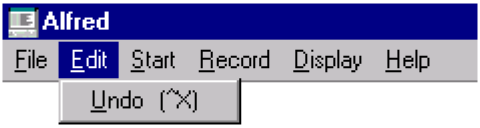
Read/Write opens the current file I/O. You must choose this option if you want to change the contents of the data file. You can also open the file I/O with the "RW" toolbar button:

Read Only returns the current file to read-only mode. This mode ensures that changes will not be made permanent (written) in the file. By default, data files are opened in read-only mode. You can also change the input status to read-only with the "RO" toolbar button:

Print sends the entire current record to the printer.

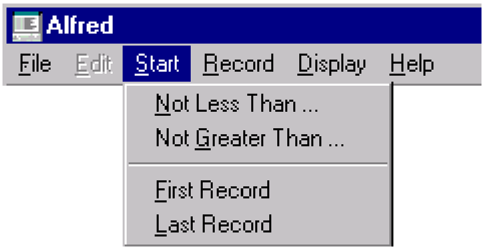
Exit shuts down the editor.

1.6.5 The Edit menu



Undo causes the last change to be undone. This menu option is available only when the data file is open for I/O and a change has been made.

1.6.6 The Start menu



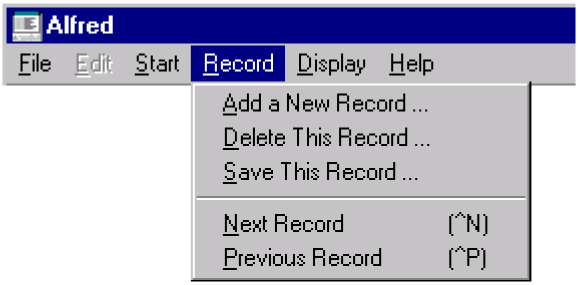
Not Less Than prompts for a key value and instructs **alfred** to display the first record whose key value is equal to or greater than the value provided.

Not Greater Than prompts for a key value and instructs **alfred** to display the first record whose key value is equal to or less than the value provided.

First Record instructs **alfred** to display the first record in the file.

Last Record instructs **alfred** to display the last record in the file.

1.6.7 The Record menu



Add a New Record prompts for a key value and allows you to enter a new record.

Delete This Record deletes the current record. A dialog box is displayed to confirm the delete action.

Save this Record saves the current record.

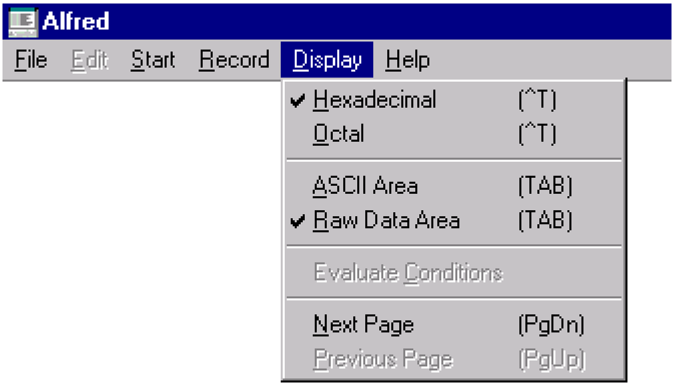
Next Record retrieves the next record. You can also display the next record by clicking on the Next Record button on the toolbar:

or by holding down the "Control" key and typing the letter "n".

Previous Record retrieves the previous record. You can also display the previous record by clicking on the Previous Record button on the toolbar:

or by holding down the "Control" key and typing the letter "p".

1.6.8 The Display menu



Hexadecimal causes the left portion of the screen (binary area) to be displayed in hexadecimal. This option is valid only when you are editing without an XFD.

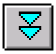
Octal causes the left portion of the screen (binary area) to be displayed in octal. This option is valid only when you are editing without an XFD.

ASCII Area moves the cursor to the right portion of the screen. This option is valid only when you are editing without an XFD.

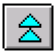
Raw Data Area moves the cursor to the left portion of the screen. This option is valid only when you are editing without an XFD.

Evaluate Conditions causes the display of data associated with a WHEN directive to be updated to match a new controlling value. Choose this option after you change a field that is the controlling value for a WHEN directive. See the *AcuCOBOL-GT User's Guide*, section 5.3.2, "Using Directives," for more information about directives.

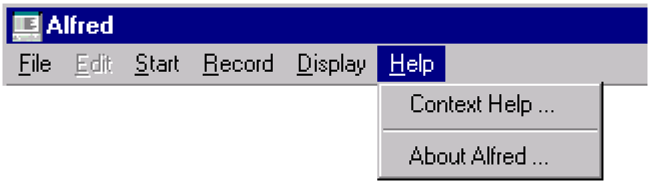
Next Page displays the next screen of data.



Previous Page displays the previous screen of data.



1.6.9 The Help menu



The **Help** menu option provides summary information about **alfred** functions.

1.7 Configuration Variables

Some aspects of **alfred** can be controlled with configuration variables. Like other ACUCOBOL-GT configuration variables, **alfred** configuration variables can be included in your regular configuration file, or in a separate configuration file, or they can be set in the environment. By default, when ACUCOBOL-GT is installed, **alfred** is set up to use a unique configuration file, named "alfred.cfg", located in the "tools" subdirectory. To use another configuration file, name the file with the "-C" option when you start **alfred**.

Or, if you start **alfred** with the program icon you'll need to redefine the icon's COMMAND LINE property (the TARGET field of the SHORTCUT property page in Windows 98) to specify the name and location of the configuration file that you want to use. The default COMMAND LINE property definition is:

```
c:\ACUCBL32\BIN\wrun32.exe -C ..\tools\alfred.cfg alfred
```

The specification of **alfred** configuration variables conforms to the same format and usage rules that apply to all other ACUCOBOL-GT runtime configuration variables (see Appendix H in Book 4, *Appendices*).

alfred's specific configuration variables include:

1.7.1 ALFRED-MESSAGE-FILE

Holds the path and name of the **alfred** message file.

1.7.2 Variables to control color:

The following eight variables allow you to customize some of the color and video attributes that **alfred** uses to display select program and record elements. The default value of each is given after the variable name.

ALFRED-LABEL-COLOR	low cyan
ALFRED-ENTRY-COLOR	low cyan
ALFRED-BANNER-LABEL-COLOR	low green
ALFRED-BANNER-TEXT-COLOR	low green
ALFRED-HIGHLIGHT-COLOR	low green
ALFRED-DISABLED-COLOR	low cyan
ALFRED-HELP-COLOR	low green
ALFRED-KEY-COLOR	high red

You can set a variable to a combination of color and video attributes by stringing the text values together, or by adding their numeric values together. For example, you can set a variable to:

```
    underline high red on black
```

or you can set it to:

```
    12325
```

Note: "12325" equals "8192 + 4096 + 32 + 5", which is interpreted as "Underline + high intensity + background black + red". Either way the values are given, you get the same effect.

The following is a list of valid color and video attributes values.

Color	**Foreground**	**Background**
Black	1	32
Blue	2	64
Green	3	96
Cyan	4	128
Red	5	160
Magenta	6	192
Brown	7	224
White	8	256

Value	**Video Attribute**
1024	Reverse video
2048	Low intensity
4096	High intensity
8192	Underline
16384	Blink
32768	Protected
65536	Background low-intensity
131072	Background high-intensity

See Chapter 9, "Color Mapping," in *AcuCOBOL-GT User Interface Programming*, for a discussion of the configuration and control of color.

1.7.3 ALFRED-ENTRY-FIELDS

This variable controls whether a box is placed around the entry field created for each field of a record. Note that entry fields are used only when you are editing a file that has an XFD (otherwise entry fields are not used). The default setting is BOXED, which indicates to place a box around all entry fields. If any other value is given, entry fields are *not* boxed.

1.7.4 ALFRED-TOOLS-FILE

This variable holds the path and name of the bitmap file to use with the **alfred** toolbar. It is predefined to point to the standard toolbar bitmap file in “..\tools\alftools.bmp”. If you want to substitute your own toolbar icons, simply redefine this variable to hold the path and name of your bitmap file. The file must contain at least six bitmaps. If this variable is blank, or if **alfred** cannot open the named file, or it does not contain bitmap data, the toolbar will use text buttons.

1.7.5 Toolbar bitmap variables

The following six variables allow you to specify which bitmap (in the bitmap file) to use with a given button.

ALFRED-INPUT-BITMAP	1
ALFRED-IO-BITMAP	2
ALFRED-PG-DN-BITMAP	3
ALFRED-PG-UP-BITMAP	4
ALFRED-PREV-REC-BITMAP	5
ALFRED-NEXT-REC-BITMAP	6

1.7.6 ALFRED-BITMAP-FILE

This variable holds the path and name of the “**alfred** by Acucorp” bitmap.

1.7.7 ALFRED-GET-PRINTER

This variable holds the name of a COBOL program that returns information about the printer. **alfred** calls the named program with a single USING argument. The program should return the name of the printer to be used and the number of lines allowed per page. You must provide the COBOL program. The group item that is passed is defined in the file “alfred.ws” located in the ACUCOBOL-GT installation subdirectory “tools”. It is currently defined as follows:

```
01  alfred-get-printer-group.
    03  printer-name     pic x(80)
    03  lines-per-page   pic 999.
```

1.7.8 ALFRED-PRINTER-NAME

This variable holds the name of the printer to use. Its value is used if ALFRED-GET-PRINTER is undefined or does not return a valid printer name. Note that if ALFRED-GET-PRINTER does not return a printer name and this variable is undefined, **alfred** will attempt to use “-P SPOOLER”.

1.7.9 Print header variables

These variables are used to configure up to nine print headers. It is expected that header definitions will frequently consist largely of text; however there is a set of codes that can be used to insert dynamic information, such as the date or time. Each code must be preceded with the percent character (“%”), followed by a code letter.

Code letter	**Number of characters used**	**Meaning**
m	2	month (01 - 12)
y	2	year
c	2	century
d	2	day of month (01 - 31)

Code letter	**Number of characters used**	**Meaning**
D	10	full date (equivalent to %c%y/%m/%d)
H	2	hour (01 - 24)
M	2	minute (00 - 59)
S	2	second (00 - 59)
T	8	full time (equivalent to %H:%M:%S)
a	3	abbreviated weekday (Sun, Mon, Tue, etc.)
h	3	abbreviated month (Jan, Feb, Mar, etc.)
r	10	full time, 12 hour, with AM or PM as necessary
p	3	page number
f	80	file name

The names of the print header variables are:

ALFRED_PRINT_HEADER_1
ALFRED_PRINT_HEADER_2
ALFRED_PRINT_HEADER_3
ALFRED_PRINT_HEADER_4
ALFRED_PRINT_HEADER_5
ALFRED_PRINT_HEADER_6
ALFRED_PRINT_HEADER_7
ALFRED_PRINT_HEADER_8
ALFRED_PRINT_HEADER_9

Index