JYACC FORMAKER Author's Guide

Contents

1 Introduction

The JYACC FORMAKER authoring utility is a full-screen editor that enables you to create and test screens for your application quickly and easily. This chapter describes that utility in detail. It is organized by function, for easy reference; Section 6 describes how to create certain types of screens.

Note on function key names

JYACC FORMAKER is available on many different computers with a variety of terminals. Because the package is terminal-independent, this manual refers to function keys by generic names (TRANSMIT, EXIT, PF2, etc.). The actual keys you use will depend on your keyboard; see the Introduction and Section 2.4.

1.1 On-line Help

The JYACC FORMAKER authoring utility features on-line help. Every pop-up window and menu has a help screen, and where appropriate each field or selection within the window will have its own. Windows with more than one screenful of information may be scrollable, or they may contain a menu of subtopics in more windows.

Pressing the HELP key will get you the most specific help available: the help window for the field under the cursor, if it has one, or for the whole pop-up. Pressing the FORM HELP key will always bring up the pop-up's help window.

Finally, in the screen editor's draw mode, pressing either HELP or FORM HELP will bring up a long introduction to the screen editor.

2 Entering and Exiting the Screen Editor

2.1 Entering the Screen Editor

You invoke the JYACC FORMAKER authoring utility by entering

xform <formname>

The optional argument <formname> is the name of an existing screen, or of a screen to be created.

The utility will respond:

```
JYACC FORMAKER Rel 4.0 Copyright (C) JYACC, Inc. 1988
Editing the form "formname".
Hit the <EXIT> key to abort editing this file,
or any other key to continue.
```

If you hit a key other than the EXIT key, JYACC FORMAKER brings up the screen. If the screen already exists, JYACC FORMAKER displays it on the screen, in test mode; otherwise, you get a blank screen in draw mode.

The mode (draw or test) can be changed by means of the PF2 function key, which acts as a toggle.

If you enter xform without supplying a screen name, or hit the EXIT key at the first prompt, the window shown in figure 1 will appear.

To create or edit a screen, enter its name and hit TRANSMIT. If you hit the EXIT key instead the utility will exit, unless it was invoked with more than one name, as

xform form1 form2 form3

The utility responds to such an invocation with:

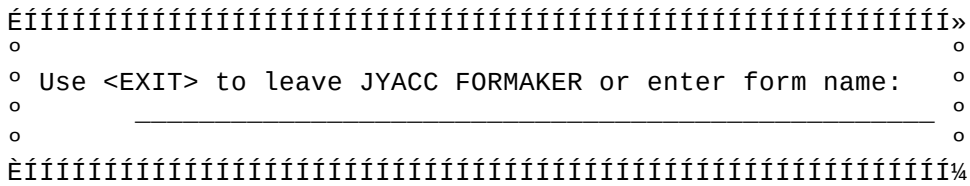

Figure 1: JYACC FORMAKER Entry Screen

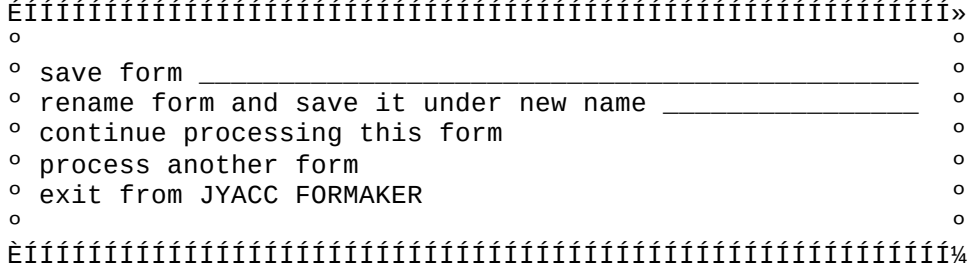

Figure 2: Exit Options Menu

JYACC FORMAKER Rel 4.0 (C) Copyright JYACC, Inc. 1988
Editing the form "form1".
Number 1 of 3.
Hit the <EXIT> key to abort editing this file,
or any other key to continue.

Hitting the EXIT key will cause the above screen to be repeated, with the next screen name and its sequence number in the list. Only after the last screen name is displayed will hitting the EXIT key produce the screen name/exit window (Figure 1). Thus, if you wish to exit the utility before reaching the end of the list, it may be more convenient to hit some other key and then go through normal exit processing, described in the following section.

2.2 Exiting the Screen Editor

You invoke the screen editor's exit functions by hitting the EXIT key when no pop-up windows are displayed. The exit options menu (Figure 2) will appear, with save shown in reverse video. To make a selection, either:

1. Use the tab, backtab, space bar, backspace, or arrow keys, to position the reverse video area to the desired option, then hit TRANSMIT, or
2. Hit the initial letter of the desired option (e.g. c for continue).

Note that choosing exit or hitting the EXIT key will exit the screen editor without saving the screen. To rename the screen:

1. Choose rename. The cursor will be positioned to a field on the right of the option line, where the name is to be entered.
2. Enter the new name.
3. Hit TRANSMIT to save the screen under its new name (or hit the EXIT key to abort, and choose another option).

To process another screen, choose process. If a list of screens was entered originally, and not all of them have been edited, the utility will re-display its initial screen with the next screen name and its number in the list. Otherwise, the screen name/exit window will appear.

2.3 Screen Editor Processing Levels and Modes

The JYACC FORMAKER screen editor operates on its top level at all times except

- while one or more pop-up windows are displayed;
- while you are moving or copying a field or display area;
- during exit functions.

When the screen editor is on its top level, all its function keys are available. At other times, none are available (except PF7 and PF8, which are used for terminating the move and copy functions, respectively). On the top level, the TRANSMIT key compiles the screen, and the EXIT key invokes exit functions. When the pop-up windows are active, the TRANSMIT key is used to effect changes, and the EXIT key is used to abort the changes, or to terminate processing on that level.

On the top level, the screen editor has two modes of operation, draw mode and test mode. You create and modify screens primarily in draw mode; in test mode, you can quickly check whether they behave as you intend. The display's status line always indicates which mode you are in, so you can check at a glance.

- In draw mode, you can enter data anywhere on the screen, using arrows and the RETURN key to position the cursor. The display shows an exact image of the screen as it is developed. Constant data (such as borders, field labels, and initial field contents) have their true colors and display attributes, as supported by the display hardware. Fields you have defined for data entry appear in draw mode as underlined areas.
- Test mode enables you to try out the screen, to see whether it works as planned and is comfortable to use. In test mode, data may be entered only in previously defined fields, and all data restrictions and field editing rules apply.

Certain function keys are used directly for toggling the mode, and for deleting, moving, and copying areas of the screen. Other function keys bring up menus and windows for entering detailed information about screen size, display attributes, data restrictions, and editing rules. The next section summarizes those keys.

2.4 Data Entry

In the JYACC FORMAKER run-time system, data entry is permitted only in fields. When you type a normal data character, it is copied into the field under the cursor, subject to certain restrictions and rules. There are also specially defined keys that move the cursor, clear areas, scroll, invoke help, etc. In this section we explain rules that apply generally to all fields. Data entry restrictions that can be applied to specific fields are explained later on, in Section 5.

Menus

Data entry rules for menu screens are completely different and much simpler than for data-entry screens, to which the rest of this section applies. In a menu there is a reverse-video cursor, often referred to as a "bounce bar," that occupies all of the current menu item. The TAB, right and down arrows, space, and RETURN keys all move the bounce bar to the next eligible menu item; the BACKTAB, left and up arrows, and BACKSPACE key all move it to the previous item. Pressing the TRANSMIT key causes the item under the bounce bar to be selected and returned to the function processing the menu. Typing enough characters to uniquely identify a menu item will cause that item to be selected. All these

behaviors are subject to modification by the library functions sm_mp_option and sm_mp_string .

Insert and Overstrike Modes

At any given time, either insert or overstrike mode is active. In overstrike mode, characters you type simply replace any previously existing data in a field. In insert mode, the old field contents are shifted, left or right according to the field's justification, to make room for the new character. The INSERT key toggles this mode.

Character Edits

Fields can be marked to permit the entry of only certain characters; for instance, a field that is to contain a number will accept only digits, and beep if you attempt to enter anything else. See Section 5.5 for full details.

Special Data Editing Keys

The following table describes the behavior, in test mode and applications, of the data editing keys defined by JYACC FORMAKER. Where their behavior in draw mode is different, the differences are noted. Where there are run-time options that modify the behavior, they are also noted.

Short	Long name	Description name
ABORT	ABORT	Causes keyboard input functions to return to their callers as quickly as possible, and sets a global flag.
BACK	BACKTAB	Move the cursor to the unprotected field with the next lower number. Validation is inhibited by default. BKSP
	BACKSPACE	Delete the character to the left of the cursor, and shift all characters to the right (left) of the cursor left (right) 1 position, according as the field is left (right) justified.
CLR	CLEAR ALL	Clear all unprotected fields in the current window, both onscreen and off. System date and time fields are updated.
DARR	DOWN	Move to the entry point of the next field below the current line, unless inhibited by sm_ok_options. If in the last line of a scrolling array, scroll up this and any parallel arrays. In draw mode, move the cursor down one line.
DELE	DELETE CHAR	Delete the character under the cursor and shift the field contents on its right (or left) one position to the left (or right), according as the field is left (or right) justified. In draw mode, everything is treated as left-justified.
DELL	DELETE LINE	Move all array occurrences below the current one in this and any parallel arrays up one line, overwriting the one under the cursor. In draw mode, delete the current line and scroll up those below.
EMOH	LAST FIELD	(HOME backwards.) Move the cursor to the beginning of the last unprotected field.
EXIT	EXIT	Cease the current operation and usually close the current window without making the changes that have been specified since the window opened.
FERA	ERASE	In a left-justified field, clears from the cursor to the end of the field; in a right-justified field, always erases the whole field. In a system date or time field, updates the value.
FHLP	FORM HELP	Display the help window for the current screen, regardless of the field in which the cursor is placed.

HELP	HELP	Display the help screen for the field in which the cursor stands, or for the screen if the field has none.
HOME	HOME	Move the cursor to the entry point of the first unprotected field. If no field is unprotected, move to the top left-hand corner of the screen.
INS	INSERT CHAR	Toggle the insert/overstrike mode of data entry.
INSL	INSERT LINE	In an array and any parallel arrays, move the current item and all below it down by one, and clear the current item. Fails if the last item in the array or one parallel is already filled. In draw mode, move the current line and all below it down by one, and blank the current line.
LARR	LEFT	Move the cursor one position to the left within a field; from the first position, move into the previous field, unless inhibited by ok_options. In draw mode, move the cursor 1 character to the left.
LP	LOCAL PRINT	Send what is in the screen buffer to a file for printing.
NL	RETURN	Move the cursor to the first unprotected field below the current line, wrapping to the top if there are none below. If in the last line of a scrolling array, scroll up this and any parallel arrays. In draw mode, move the cursor to the beginning of the next line, wrapping at the bottom.
RARR	RIGHT	Move the cursor one position to the right. If at the right end of a shifting field, shift in offscreen data. At the very end of the field, move to the beginning of the next unprotected field. May be inhibited, equated to TAB, or otherwise altered by ok_options. In draw mode, move the cursor right by one position.
REFR	RESCREEN	Refresh the screen if it gets scrambled by line noise or output from other programs.
SPGD	PAGE DOWN	In a scrolling array, scroll it and any parallel arrays down by several lines.
SPGU	PAGE UP	In a scrolling array, scroll it and any parallel arrays up by several lines.
TAB	TAB	Move the cursor to the beginning of the next unprotected field. If in the last such field, move to the first such field. Field entry and exit routines are performed as appropriate; if field validation fails, the cursor remains in the current field.
UARR	UP	Move the cursor to the entry point of the previous field above the current line, unless inhibited by ok_options. If in the first line of a scrolling array, then scroll down this and parallel arrays, if any, if any previous items exist. In draw mode, move the cursor up one line.
XMIT	TRANSMIT	Make effective any choices made in the current window, frequently closing the window as a side effect. Causes validation of the entire screen. In draw mode, convert all underscored strings to fields.
ZOOM	ZOOM	Expand the scrolling and/or shifting field under the cursor into a pop-up window, where more of the field will be visible. You may enter data into this window just as into the field itself.

Special Function Keys

The table below describes the actions of program function keys in the screen editor.

Key	Description
PF2	Toggles the utility between draw mode and test mode.
PF3	Brings up the form characteristics window.
PF4	Brings up the field characteristics window, if the cursor is positioned within a field, or the display attributes window if the cursor is positioned within a display area.
PF5	Brings up the field summary window.
PF6	Deletes the display area, field, or array of fields within which the cursor is positioned.
PF7	Moves the display area, field, or array under the cursor.
PF8	Copies the display area, field, or array under the cursor.
PF9	Repeats the last move, copy, delete, graphics, or change field characteristics sequence. See below for details.
PF10	Brings up the graphics character selection window.

The PF9 key in the screen editor repeats, internally, the last sequence of keystrokes:

1. starting with PF4 and ending with the closing of the field characteristics (or display attributes) window, or
2. consisting of PF6 (only), or
3. beginning and ending with PF7 (move sequence), or
4. beginning and ending with PF8 (copy sequence), or
5. starting with PF10, the graphics key, and ending with TRANSMIT.

The screen display is changed only to reflect the end result of the sequence. You must be careful to position the cursor before hitting PF9 so that the repeated sequence makes sense. In particular, a sequence that changes field characteristics should be repeated only when the cursor is within a field; a sequence that changes the attributes of a display area should be repeated only in another display area.

3 The Screen as a Whole

Screens created by JYACC FORMAKER are, by default, as big as your display. Smaller screens are convenient for use as windows, which temporarily overlay part of a larger screen. Screens can be outlined by a reverse-video or graphics border. On color displays, you can select a border color; on displays that support background color, you can select a background color as well.

For quicker and more consistent creation of screens, you can define several draw-field symbols, and an initial display attribute for display data. These defaults, as well as display text and any fields common among screens in your application, can be stored in a template and used as a basis for the creation of many screens.

Finally, you may define a help window to provide basic information to a user of your screen, and screen-entry and screen-exit functions to perform initialization and wrap-up on the screen. xform provides access to all these screen characteristics through the screen characteristics window (Figure 3), which you can call up using PF3.

3.1 Screen Size

To change the screen size:

1. Hit the PF3 key to bring up the screen characteristics window (Figure 3). The window will show the current size.
2. Enter the new number of lines (or TAB, if no change).
3. Enter the new number of columns (if changed).

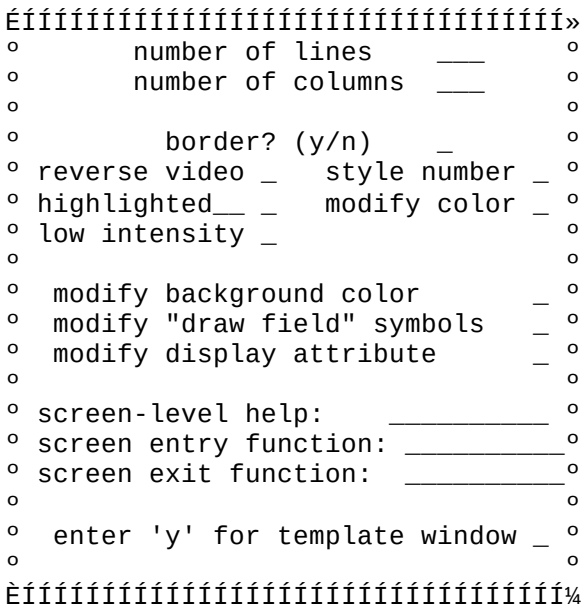

Figure 3: Screen Characteristics

4. Hit TRANSMIT to effect the change (or EXIT to abort).

xform will not allow a screen to be made larger than the display, nor smaller than its contents, so it may first be necessary to reduce the size of a screen by deleting or rearranging some of the fields.

3.2 Borders

To add or change a border:

1. Hit PF3 to bring up the screen characteristics window (Figure 3). It will show the current border characteristics; if there is no border, several of the fields shown in the figure will not be visible.
2. After border? enter y if a border is desired, or n for no border.
3. If a border is chosen, the window will display some additional yes/no prompts:

    a. reverse video (defaults to y).
    b. highlighted (defaults to y). On displays that support low intensity but not highlighting, the prompt will be low intensity instead.
    c. style number (defaults to 0). Enter a digit from 0 through 9. The chosen border will immediately appear around the screen characteristics window, and the cursor will remain in this field so you can try more selections. (The actual styles available depend on your video file.)
    d. modify color. See the next section.

4. Hit TRANSMIT to effect the change (or EXIT to abort).

3.2.1 Border Colors

On color displays, the screen characteristics window has an option for border color selection (Figure 3). To change the color of a border:

1. Hit PF3 to bring up the screen characteristics window.
2. Enter y after border.

Figure 4: Color Selection Menu

3. Tab to modify color and enter y. The color selection menu (Figure 4) will appear, with the current color shown in reverse video. (The default color is white.) To change the color, either:

   a. Using TAB, BACKTAB, space, BACKSPACE, or the arrow keys, position the reverse video area to the desired color, and hit TRANSMIT, or
   b. Hit the initial letter of the desired color (r for red, etc.). Choosing exit or hitting the EXIT key will close the window without changing the border color.

At this point, the screen characteristics window will still be displayed, and you can make any other desired changes. When finished, hit TRANSMIT to save the new screen characteristics, or EXIT to abort. The screen's border, if any, will now have the color that was selected on the color menu.

3.3 Background Color

If background color is available, the screen characteristics window will display an option for background color selection (Figure 3). To change the background color:

1. Hit PF3 to bring up the screen characteristics window.
2. After modify background color enter y. The color selection menu (Figure 4) will appear, with the current color shown in reverse video. The default is no color (black). To change the color, either:

   a. Using TAB, BACKTAB, space, BACKSPACE, or the arrow keys, position the reverse video area to the desired color, and hit TRANSMIT, or
   b. Hit the initial letter of the desired color (r for red, etc.). Either choosing exit or hitting the EXIT key will close the window without changing the background color.

At this point, the screen characteristics window will still be displayed, and you can make further changes if you like. When you are done, hit TRANSMIT to save the new screen characteristics, or EXIT to abort. The screen's background will now be the color that was selected on the color menu.

3.4 Draw-field Symbols and Field Defaults

The underscore character is used by default for drawing fields. The resulting fields, when first compiled, are underlined and highlighted (if highlighting is available), and have no character or field edits. You can change those field

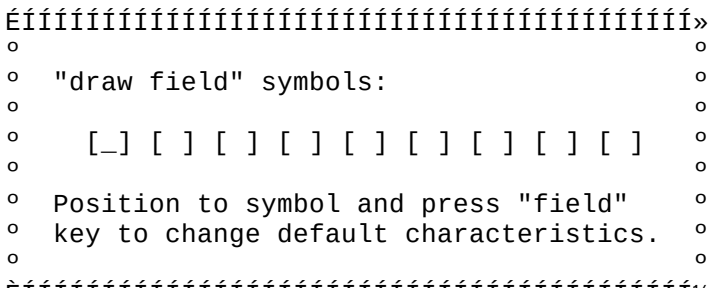

Figure 5: Draw-field Symbol Window

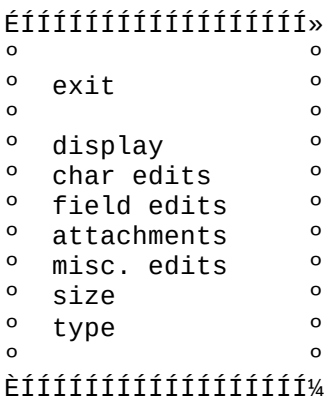

Figure 6: Default Field Characteristics Menu

characteristics, and define more draw-field characters for creating fields with different initial characteristics. To specify draw field symbols or initial field characteristics:

1. Hit PF3 to bring up the screen characteristics window (Figure 3).
2. Enter y after modify "draw field" symbols. The draw-field symbol window (Figure 5) will be displayed. Initially the window will show only one draw-field symbol, the underscore; you can define up to 9 symbols.
3. Change or add draw-field symbols by entering them in the window.

To change the initial field characteristics, position the cursor to the corresponding draw-field symbol, and hit the PF4 key to bring up the default field characteristics menu (Figure 6). This menu is just like the regular field characteristics menu, Figure 9; refer to Section 5 for its use, as well as explanations of the many different characteristics you can give your fields.

Note that different symbols can be used to create fields with different initial characteristics. The initial characteristics as specified above will apply, however, only to fields that are created afterwards; previously created fields will not change. If you want to use the same draw-field symbols for several different screens, you may find it convenient to define the symbols, as above, for one screen, and then use that screen as a template for others; see Section 3.8.

3.5 Default Display Attributes

It is also possible to change the attributes with which display data is first written to the screen, thus:

1. Hit PF3 to bring up the screen characteristics window (Figure 3).

2. Enter y after modify display attribute. The display attributes window (Figure 8) will appear, with the current attributes marked with a y.
3. To turn on an attribute enter y; to turn it off, enter n or space. More than one attribute may be selected.
4. If you have a color display, the initial color can be changed by entering y after modify color, and proceeding as in Section 4.2.1.
5. Hit TRANSMIT to effect the change, or EXIT to abort. Note that a color selection, if any, will take effect even if the other attribute changes are aborted.

Display data created after this point will have the display attributes specified above; display data already on the screen will retain their old attributes.

3.6 Help Windows on a Screen Basis

Note: the remainder of Section 3 uses some features of JYACC FORMAKER that have not yet been presented. If you are reading this manual for the first time, you may want to skip to the beginning of Section 4.

A help window may be specified on a screen basis. The screen so designated will be displayed as a window whenever the FORM HELP key is pressed. It will also be displayed when the HELP key is struck and the cursor is not in a field that has its own help or item selection window.

To specify a help window for a screen:

1. Hit PF3 to bring up the screen characteristics window (Figure 3). If a help window has already been specified for this screen, its name will be displayed.
2. After screen-level help enter the name of the screen that is to serve as a help window for the current screen. You may optionally follow the window name with the line and column at which it is to appear, as helpwin(5,25).
3. Hit TRANSMIT to effect the change, or EXIT to abort.

The creation of help windows is described fully in Section 6.2.

3.7 Screen-entry and Exit Functions

These functions are analogous to the field-entry and exit functions described later. They are called by library functions such as r_window, d_form, or close_window at the very beginning and end of a screen's active life. They may be used to initialize the screen, to allocate or release resources, or to do whatever the application needs at that point. Refer to the Programmer's Guide for details. To specify such a function, follow the instructions below. For simplicity, they refer only to screen-entry functions, but the procedure for screen-exit functions is exactly analogous.

1. Press PF3 to bring up the screen characteristics window (Figure 3).
2. Tab to the field labeled screen entry function and type in the function name.
3. Hit TRANSMIT to accept the screen-entry function, or EXIT to discard the change.

Functions written in a standard programming language must be registered with the JYACC FORMAKER library through a call to install; see the Programmer's Guide. You can also use a procedure written in JPL, the JYACC Procedural Language, and stored in a file by entering jpl filename in the screen-entry or exit function field.

3.8 Using Another Screen as a Template

When designing a screen, a copy of an existing screen can be used as a template. Screen templates can be particularly useful for creating several screens with

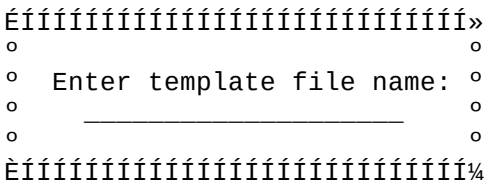

Figure 7: Template Window

similar characteristics, including initial display attributes (Section 3.5) and draw-field symbols (Section 3.4). To make use of this feature:

1. Enter the screen editor with the name of the new screen (Section 2).
2. After the screen is blanked, hit PF3 to bring up the screen characteristics window (Figure 3).
3. Tab to the last field and enter y. The window shown in figure 7 will pop up.
4. Enter the name of the existing screen and hit TRANSMIT, or EXIT to abort. The screen is now initialized to a copy of the existing screen.
5. You will be prompted to hit TRANSMIT to accept the template, or EXIT to reject it.

Loading a template destroys whatever fields or data you had previously created for the new screen.

4 Display Data

Display data include form headings, field headings, and all constant data that are displayed as part of the screen. Borders and messages to be displayed on the status line are not considered display data. To enter display data:

1. Make sure you are in draw mode. Hit the PF2 key if necessary.
2. Use the RETURN and arrow keys to position the cursor.
3. Enter the display data as you want them to appear on the screen. All displayable characters can be used, except those reserved for drawing fields. If you have not defined any draw-field symbols yourself (Section 3.4), the only reserved character is the underscore.

The remainder of this section will sometimes treat an area of display data as a unit. Such a display area consists of display data that are on one line; not interrupted by the start or end of a field; and not separated by more than one blank.

Entry of display data follows the rules defined in Section 2.4. When a key pays attention to the extent of a field, a display area is treated like a field.

4.1 The Graphics Selection Window

If you need to enter special graphics characters, hit the PF10 key to bring up the graphics selection window. JYACC FORMAKER will display all the graphics characters defined for your terminal in a window. Use arrow keys to position the cursor over the character you want, hit TRANSMIT, and it will be placed on your screen at the current cursor position. To insert a character repeatedly, use the PF9 (repeat) key instead of bringing up the window again.

4.2 Display Attributes

You can change the appearance of a display area by changing its display attributes, thus:

1. Position the cursor anywhere within the display area.

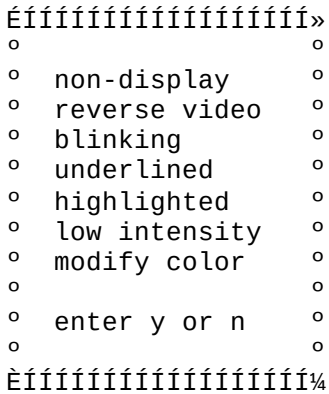

Figure 8: Display Attributes

2. Hit the PF4 key to bring up the display attributes window (Figure 8). It will show which attributes are currently in effect. The list of available attributes will depend on the video file associated with the terminal, but should always include non-display and underlined. (On displays that cannot underline, JYACC FORMAKER will simulate underlining by placing an underscore character wherever the actual character is a blank.)
3. To set an attribute, enter y; to reset one, enter n or space. Attributes may be combined.
4. Hit TRANSMIT to effect the change, or EXIT to abort.

To change the display attributes of a field, see Section 5.4.

The above procedure will change the attributes of a display area after it has been created. To change the default attributes of display data, see Section 3.5.

4.2.1 Color

On screens that support color, the display attributes window has an additional option for color selection (Figure 8). To change the color of a display area:

1. Position the cursor anywhere within the display area.
2. Hit the PF4 key to bring up the display attributes window.
3. Enter y after modify color. The color selection menu (Figure 4) will be brought up on the screen, with the current color shown in reverse video.
4. To change the color, either:

    a. Using the TAB, BACKTAB, space bar, BACKSPACE, or arrow keys, position the reverse video area to the desired color, and hit TRANSMIT, or
    b. Hit the initial letter of the desired color (r for red, etc.). Either choosing exit or hitting the EXIT key will close the window without changing the color.

At this point, the display attributes window will still be displayed; you may continue to change attributes as desired. When finished, hit TRANSMIT to set the display attributes (or EXIT to abort). The color will remain as selected on the color menu.

4.3 Deleting, Moving, and Copying Display Data

Individual display characters can be deleted, in draw mode, with the DELETE CHAR key (Section 2.4). Similarly, an entire display area or the end of a display area can be deleted by using the ERASE key in draw mode. An entire display area

can also be erased by positioning the cursor anywhere within it and hitting the PF6 key.

You can reposition or copy a display area anywhere within a screen. If the new location overlays a field, the display area becomes initial data for that field. It is possible to overlay (and lose) other display data. The procedures are identical, except that you use PF7 to move and PF8 to copy:

1. Position the cursor anywhere within the chosen display area.
2. Hit the PF7 (PF8) key. The extent of the data to be moved will be shown by brackets. If possible, all fields will appear underlined, and non-display fields will be made visible.
3. Position the brackets to the display data's new location on the screen, using the arrow, TAB, BACKTAB, PAGE UP and PAGE DOWN keys. (The arrow keys move the data one line or column at a time. The TAB and BACKTAB keys move the data ten columns at a time, wrapping past the sides of the screen. The PAGE UP and PAGE DOWN keys move the data five lines at a time, wrapping past the top and bottom of the screen.)
4. Hit the PF7 (PF8) key again to deposit the display area in its new location, or hit TRANSMIT. Hit the EXIT key to abort the procedure.

5 Fields

Fields are areas you define for data entry when the screen is used. Each field must be contained within a single line of the screen. Using pop-up menus, you can set display attributes separately for each field, restrict what may be entered into fields, and specify many other field characteristics.

5.1 Creating Fields and Compiling the Screen

To create a field:

1. Make sure you are in draw mode. Hit the PF2 key if necessary.
2. Use the RETURN and arrow keys to position the cursor.
3. Type underscores (or any key defined as a draw-field symbol, Section 3.4) to define the extent of the field.

At this point, the field is only display data, and all rules for modifying, deleting, moving, and copying display data apply. To change areas defined by the underscore (or other symbol) into actual fields, hit the TRANSMIT key to compile the screen. When a screen is compiled, any areas containing underscores or other draw-field symbols and not within existing fields or borders are converted into fields. Fields are then renumbered, from top to bottom and from left to right within each line. Compilation is fast, and may not be noticeable. A screen is automatically compiled:

1. When the PF2 key is used to toggle from draw mode to test mode.
2. Whenever the TRANSMIT key is struck while the screen editor is at its top level (Section 2.3).
3. Before xform's exit functions are performed (when the EXIT key is struck at the top level).
4. After a field is deleted, moved, or copied.
5. After a field's characteristics have been changed.

Only the first three actions in the list will cause new fields to be created.

Initial data may be entered into a field in test mode, in which case all field restrictions apply, or in draw mode, in which case anything may be entered. Initial data may also be moved or copied onto a field from a display area; see Section 4.3. When a screen is saved, data contained in a field are saved with the screen, and will be displayed as the field's initial value when the screen is displayed at run-time. There is one exception: contents of date and time fields that default to system values are not saved (see Sections 5.8.2 and 5.8.2).

To modify field characteristics, including the size of a compiled field, press the PF4 key with the cursor located anywhere within the field. (If the field is protected against tabbing, you will not be able to move the cursor there in test mode; use the PF2 key to get into draw mode, in which all fields are accessible.) The field characteristics menu, shown in Figure 9, will pop up near your field. Each option on the menu is described in great detail in subsequent sections.

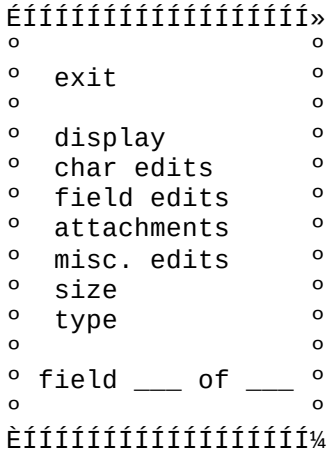

Figure 9: Field Characteristics Menu

5.2 Deleting, Moving, and Copying Fields

The PF6, PF7, and PF8 keys will delete, move, and copy fields as well as display data (see Section 4.3). A field can be moved or copied anywhere within a screen, as long as the new location doesn't overlap a side border or another field. Note that it is possible to overlay, and thus lose, display data.

When a field is deleted, moved, or copied, the field's current contents go with it. When a field is moved, it retains its display attributes and restrictions on data. When a field is copied, its display attributes and restrictions on data are copied also. The sole exception to this rule is that field names (Section 5.7.1) are not copied. An array of fields (Section 5.3.2) is always deleted, moved, or copied as a unit.

Since the data modification keys (Section 2.4) are applied to data entered within a field, they cannot be used to change the extent of the field itself. Changing the length of a field is described in Section 5.3.

5.3 Field Size

The simplest sort of field you can define with JYACC FORMAKER is just part of a single line on the screen. Every field takes up at least one space on the screen; but fields can be made much larger, larger even than the physical display. You can extend a field horizontally by making it shifting; then only a portion of the field's contents will be visible on the display. You can extend a field either vertically or horizontally by making it an array, and you can expand an array offscreen by making it scrolling. A field's size, then, has four parameters:

1. its visible, onscreen length;
2. its total length, which is greater if it's shifting;
3. its number of onscreen array elements;
4. its total number of data items, which is larger if the array is scrollable.

Here are definitions of a few more terms that occur frequently in the following pages.

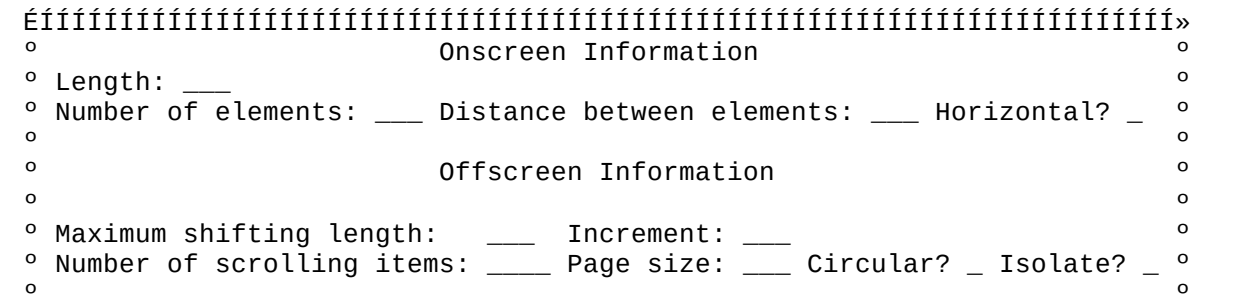

Figure 10: Field Size Window

Each of the fields that comprise an array is referred to as an element of the array. Each element occupies a fixed position on the screen.

If a field, or an array of fields, is scrollable, each individual entry is referred to as an item of data. Suppose an array of 3 elements is made scrollable, with a maximum of 10 items. When you begin entering data, item 1 is written to element 1, item 2 is written to element 2, and item 3 is written to element 3. After the third item is written, scrolling moves the first item to an offscreen buffer, moves item 2 into element 1, and moves item 3 into element 2; you then proceed to enter item 4 into element 3.

Although the appearance on the screen is quite different, from a program's point of view it doesn't matter whether a data item is entered into a non-scrollable array of 5 elements, or a scrollable array of 2 elements that allows a maximum of 5 items. To make those differences transparent to programs, the concept of an occurrence was introduced. In a scrollable field or array, each occurrence is an item, and an occurrence can be either on the screen or offscreen. In a non-scrollable array, each occurrence is an element, and it is always onscreen. A non-scrollable field that is not part of an array consists of a single occurrence.

To modify a field's size:

1. Position the cursor anywhere within the field.
2. Hit the PF4 key to bring up the field characteristics menu (Figure 9).
3. Choose size to bring up the field size window (Figure 10. It contains the field's current size parameters.

Here is how to use the field size window. Following sections contain more detailed explanations of arrays, shifting, and scrolling.

Length	Enter the field's visible, onscreen length. If lengthening the field causes it to overlap another field, you will get an error message.
Number of elements	To make a field an array, enter a number greater than 1; the array will have this many onscreen elements. To reduce an array to a single field, enter 0 or 1. See Section 5.3.2.
Distance between elements	Enter the number of spaces between successive array elements, a number of lines (for vertical arrays) or columns (for horizontal arrays). The default value of distance is 1. See Section 5.3.2.
Horizontal?	For a vertical array, enter n (or leave blank); the new elements will appear below the field you are modifying. For a horizontal array, enter y, and the new elements will appear to its right. See Section 5.3.2.

Maximum shifting length	Enter the maximum possible length of data to be put into the field. Must be greater than the onscreen length. See Section 5.3.1.
Increment	Enter the number of characters by which the field's contents should be shifted when you arrow "beyond" its edge. See Section 5.3.1.
Number of scrolling items	Enter the greatest number of data items the field or array can possibly contain; must be greater than the number of onscreen elements. See Section 5.3.3.
Page size	Enter the number of items by which the PAGE UP and PAGE DOWN keys should scroll the field, or leave blank for default. This entry must be less than or equal to the number of onscreen elements; the default is one less, or one for a single scrolling field. See Section 5.3.3.
Circular?	Enter y if the scroll should wrap from bottom to top on down arrow, and from top to bottom on up arrow. If you enter n, those arrow keys will leave the field when the end of the scroll is reached. See Section 5.3.3.
Isolate?	Enter y if the field should not scroll when a parallel field scrolls, or n to include it in parallel scrolling. See Section 5.3.3.1.

5.3.1 Shiftable Fields

A shiftable field is one that can hold data wider than the field's onscreen length. When a character is keyed into the last position of the field, the visible contents of the field are shifted left by the field's shifting increment, a user-defined constant, and the cursor is positioned immediately to the right of the last character entered. Data can be entered up to the length specified as maximum shifting length. When the maximum size is reached, the cursor tabs to the next field (unless tabbing has been inhibited by the no-autotab option or the return entry option).

When the cursor is at the first position of the field, and the beginning of the data is not visible, the left arrow key can be used to shift the field's contents right by one shifting increment. Similarly, when the cursor is at the last position of the field, the right arrow can be used to shift the field's contents to the left. Tabbing out of the field leaves its visible contents unchanged; tabbing or backtabbing to the beginning of the field resets its contents to the beginning of the data. If the cursor is in a protected shifting field, only the data (not the cursor) will move when you press the arrow keys.

If the screen positions immediately adjacent to a shifting field contain neither display data nor other fields, shifting indicators will appear on the screen whenever there are additional data offscreen to either the left or the right of the field's visible contents.

Fields comprising a shiftable array all shift together. Tabbing into one field of such an array, therefore, will reset the contents of all the fields (since the field that has been tabbed into must be reset).

5.3.2 Arrays

An array is a set of fields that can be treated as a unit. The fields that make up an array have the same length, common field characteristics, and starting locations separated by a constant vertical or horizontal distance. If you want a horizontal array of 15-character fields with five spaces between them, the distance must be 5; however, for a vertical array with one blank line between elements, the distance must be 2.

Any change to the specifications of a field within an array changes the whole array. The PF6, PF7 and PF8 keys delete, move, and copy an array as a unit. Shifting and scrolling also apply to an array as a whole. Finally, only one field name (Section 5.7.1) can be assigned to an array; individual array elements can be referenced by field name and element number, or by field number.

Arrays can be either horizontal or vertical, not both. You can achieve a spreadsheet effect by creating a group of parallel vertical arrays (see Section 5.3.3.1).

5.3.3 Scrollable Fields

Scrollable fields are usually defined as arrays (Section 5.3.2), but individual fields can be scrolled as well. A scrollable field or array displays the visible portion of a larger set of data items. A field can be both shiftable (Section 5.3.1) and scrollable.

The first data entered go into the field or array of fields visible on the screen. When the visible fields become full, the contents of the first disappear from the screen; if it is an array, the contents of each succeeding element move up into the previous element, and the last element is again available for data entry.

If you don't want to fill each field, the same scrolling effect can be achieved by hitting the RETURN or down arrow key. The field or array will scroll as long as there is more data to display. Scrolling in the opposite direction is achieved by hitting the up arrow key, with the cursor in the first field. The PAGE UP and PAGE DOWN keys will scroll the current scrollable array (or the scrollable array closest to the current cursor position) by the number of data items given in page size in the field size window. The TAB and BACKTAB keys normally have no effect on scrolling, and can cause the cursor to leave the scrolling area; see the section on next field edits for an example of how they can cause scrolling.

When a field or an array scrolls, every field or array parallel to it will also scroll; see Section 5.3.3.1.

You may define a scrolling field as circular. When you press down-arrow with the cursor on the last item of such a scroll, it will scroll to the first item instead of exiting the field. The RETURN and up-arrow keys wrap around in the same way. To insert a new item into a circular field or array, you must use the INSERT LINE key.

5.3.3.1 Parallel Arrays

When a field or array is scrolled, any fields or arrays parallel to it will scroll too. See the preceding sections for a discussion of scrolling. Vertical (horizontal) scrolling arrays are considered parallel if

1. they start at the same line (column) of the screen;
2. the offsets between elements are the same;
3. they contain the same number of onscreen elements;
4. they have the same maximum number of scrollable items.

Single fields are considered parallel if they meet the first and last criteria (the others don't apply). Non-scrolling arrays are never considered parallel. Scrolling and shifting fields that are parallel do not shift together.

You can exclude a field that meets the above requirements from parallel scrolling by placing y in the isolate field of the field size window.

5.4 Field Display Attributes

Fields are displayed, by default, underlined and highlighted (if those attributes are available). To change the display attributes of a field:

1. Position the cursor anywhere within the field.
2. Hit the PF4 key to bring up the field characteristics window (Figure 9).
3. Choose display to bring up the display attributes window (Figure 8). The window will list the attributes supported by your display, and show which are currently in effect.
4. To turn on an attribute enter y; to turn it off, enter n. More than one may be active.
5. On screens that have color, the display attributes window has an additional option for color. To change the color of a field, enter y after modify color, and follow the procedure in Section 4.2.1.
6. Hit TRANSMIT to effect the change (or EXIT to abort).

At this point, the field characteristics window will still be displayed. You may choose exit or hit the EXIT key to close the window, or modify other field characteristics by choosing another option.

Note that in draw mode all fields appear underlined, and the contents of non-display fields are displayed. In test mode, fields are displayed with their real attributes.

The above procedure will change the attributes of a field after it has been compiled. To change the default attributes with which fields are initially created, see Section 3.4.

5.5 Character Edits

Character edits, or filters, are restrictions on what may be keyed into a field. For example, if a field is restricted to digits only, an attempt to enter a letter into the field will make the bell ring, and the letter will be discarded (unless you are in draw mode). The default edit is unfiltered, or all characters permitted. Defining a field as a currency field (Section 5.8.4) will not automatically make it numeric. To put character restrictions on a field:

1. Position the cursor anywhere within the field.
2. Hit the PF4 key to bring up the field characteristics window (Figure 9).
3. Choose char edits to bring up the character edits window (Figure 11). The current option will be shown in reverse video.
4. To change the option, either:

    a. Position the reverse video area to the desired option using the TAB, BACKTAB, space, BACKSPACE, or arrow keys, then hit TRANSMIT, or
    b. Hit the initial letter of the desired option (such as d for digits only).

Either choosing exit or hitting the EXIT key will close the window without changing the option. At this point, the field characteristics window will still be displayed; you may choose exit or hit the EXIT key to close the window, or set other field characteristics by choosing another option.

The above procedure will set the character edits of a field after it has been compiled. To set default character edits, see Section 3.4. The following character edits are available:

unfiltered	allows entry of all characters, without restriction.
digits only	allows entry of the digits 0-9 only. In a normal (left-justified) digits-only field, no spaces may remain

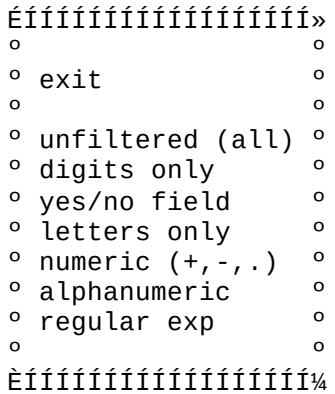

Figure 11: Character Edits Menu

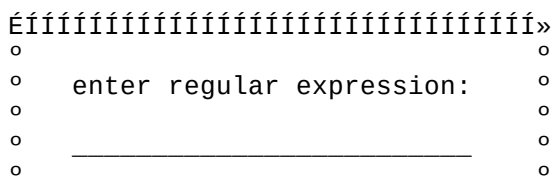

Figure 12: Regular Expression Window

blank to the left of any digit entered. In a right-justified digits-only field, no spaces may remain blank to the right of any digit entered.
A special feature of a digits-only field is that punctuation (any non-digit characters) within the field is skipped during normal data entry. The punctuation characters themselves must be entered in draw mode, or before the field is set to digits-only; they cannot be deleted in test mode. This enables you, for instance, to define a digits-only social security number field, with the initial value - - . During data entry, the dashes are permanent.

yes/no — allows entry of only the initial letters of yes and no, normally y, Y, n, N, or space (which is converted to n). This edit actually uses the initial letters of the SM_YES and SM_NO entries in the message file. The displayable characters can be further limited by setting the field edits to upper or lower case (Section 5.6).

letters only — allows entry of the letters a-z, A-Z, and the space character.

numeric — allows entry of the digits 0-9, the plus sign, minus sign, and at most one decimal point. A plus or minus sign, if present, must be leftmost in the field. In a left-justified numeric field, no blanks are allowed to the left of any character entered; in a right-justified field, no blanks may occur to the right of any character entered.

alphanumeric — allows entry of the digits 0-9, the letters a-z and A-Z, and the space character.

regular exp — allows entry according to a flexibly defined template, which you enter in the window of Figure 12. A description follows in Section 5.5.1.

5.5.1 Regular Expressions

A regular expression is a pattern or template made up of characters. It divides ordinary character strings into two kinds: those that match the pattern, and those that don't.

JYACC FORMAKER supports regular expressions in the style of the UNIX editors, and uses them to check that the contents of a field conform to a pattern. You can define the pattern in a way that is extremely flexible. Other JYACC FORMAKER character edits, such as numeric, force every character entered in a field to belong to the same type; with regular expressions, you can restrict different parts of the field to different character types, or classes.

When JYACC FORMAKER checks a field against a regular expression, it steps through the field data and the regular expression together. It matches as many field characters as it can against the first subexpression before going on to the next, and quits at the first mismatch.

Here is an example of a regular expression. This one defines a sort of ID number that is three digits, followed by a dash, followed by at least three letters or numbers, up to the length of the field:

```
[0-9]\{3\}-[a-zA-Z0-9]\{3,\}
```

If you didn't understand that, read on.

5.5.1.1 Character and Field Regular Expressions

You may install a regular expression as a character edit on a field; then each character typed into that field causes the whole field to be checked against the regular expression. If the check fails, JYACC FORMAKER beeps and rejects the character, as with simpler character edits.

If, on the other hand, you install a regular expression as a field edit, then it is not checked until you tab from the field. If that check fails, JYACC FORMAKER displays an error message and positions the cursor to the first character that failed to match.

You may combine a regular expression field edit with any character edit, including another (compatible!) regular expression. For instance, to prevent the entry of numbers with leading zeros into a field, you could make the field digits-only and give it a regular expression field edit of

```
[1-9][0-9]*
```

The non-regular character edits are, of course, more efficient at run-time.

5.5.1.2 Forming Regular Expressions

There are two kinds of rules for constructing a regular expression. One kind tells you how to form a simple expression, and the other tells you how to combine expressions into a more complex expression. The basics of regular expressions are quite simple; however, by combining them, you can quickly arrive at expressions that are quite complex. The following discussion, therefore, proceeds from simple rules for forming simple expressions to somewhat more complicated rules for combining and repeating expressions.

Simple expressions

The simplest regular expression is a single character, which matches itself: the regular expression z matches the string z. There are only a few characters that are special and do not match themselves; they are explained below. Blanks are not special, but they are not ignored either; a blank in a regular expression matches a blank in a field. (This includes leading blanks.)

A dot (.) is a special character; it matches any single character at all, including (but not limited to) itself.

The backslash (\) is also special. It is a quote character: it forces the following character to match itself, like an ordinary character, even if that character is special. For instance, the sequence \. matches a dot, and only a dot; the sequence \\ matches a single backslash. The sequence \z matches a single z; here the backslash changes nothing.

Character classes

A group of characters between brackets ([]) matches a single occurrence of any of the characters; [13579] matches any odd digit, and [aA] matches an a of either case. The group of characters is called a character class. The order of characters in a class is not significant.

Long lists of consecutive characters can be abbreviated using a hyphen (-). For instance, [a-z] matches any lowercase letter, and [A-Za-z] matches any letter at all. (Owing to the nature of the ASCII collating sequence, [A-z] matches all letters plus some punctuation characters that fall between Z and a.) You may use any number and combination of characters and ranges within one set of brackets.

You can also negate a character class, that is, cause it to match any character except those between the brackets. Do this by placing a caret (^) immediately after the left bracket. The expression [^0-9+.-] matches any non-numeric character.

Note from the previous example that special characters other than ^-] are not special in a character class, i.e. between brackets. You do not need to quote dot with a backslash to include it in a character class.

All the non-regular JYACC FORMAKER character edits can be simulated with character classes:

unfiltered	.
digits only	[0-9]
yes/no	[YyNn]
letters only	[A-Za-z]
numeric	[+-][0-9.]
alphanumeric	[A-Za-z0-9]

Concatenating subexpressions

The simplest way of combining two or more expressions is to put one after another. They then match whatever matches the first, followed by whatever matches the next, and so on. The expression JYACC matches the string JYACC; the expression a[0-9] matches a followed by a digit.

Repeating subexpressions

The star (*) causes the preceding subexpression to match zero or more characters that match the subexpression, instead of only one. The expression [0-9]* matches any number or none at all; [0-9][0-9]*, however, matches any number with at least one digit.

You can also give a more definite repeat count for an expression, enclosing it in quoted curly braces \{ and \}. The repeat count follows the subexpression, and has three possible forms:

\{n\}	exactly n repetitions
\{n,\}	n or more repetitions
\{n,m\}	at least n repetitions, but no more than m

For example, [0-9]\{5\} gives you a five-digit number, or an old-style zip code.

Repeat counts and the star are restricted to the kinds of expressions we have met so far; they may not be applied to grouped expressions, which are explained next.

Re-matching subexpressions

To re-match an expression or sequence of expressions, use quoted parentheses \(and \) around them. If you place a quoted number later in your expression, say \1, it will match whatever the first subexpression surrounded by \(\) matched. \2 will rematch the second grouped subexpression, and so on.

Note that \n does not reproduce subexpression n, but the actual character sequence that it matched earlier in the field data. The expression \([0-9]*\)\.\1 will match 123.123, or any other real number where the integer and fractional parts are the same; it will not match 123.45.

It is a confusing aspect of the backslash that it makes special characters ordinary (for purposes of matching), but also makes certain ordinary characters special (for purposes of grouping). C'est la guerre.

Some more examples

[iI][cC][Ee] matches ice, icE, iCe, iCE, Ice, IcE, ICe, or ICE.

212-[0-9][0-9][0-9]-[0-9]\{4\} matches a telephone number in Manhattan or the Bronx.

[0-9]\{3\}-[0-9]\{2\}-[0-9]\{4\} matches a Social Security number.

[a-zA-Z_][0-9a-zA-Z_]* matches an identifier in the C language.

[+-]\{0,1\}[0-9]*\.[0-9]* matches a floating point number, and [dDeE][+-]\{0,1\}[0-9]\{1,\} an exponent, in the FORTRAN language.

5.5.1.3 Summary of Special Characters in Regular Expressions

\	backslash	makes any special character, including itself, ordinary; makes the ordinary characters {}() and numbers special, in certain contexts. .
	dot	matches any single character. []
	brackets	surround a character-class subexpression. ^
	caret	immediately following [, negates the character class -
	hyphen	within brackets, denotes a character range (unless last). *
	star	causes the preceding subexpression to match zero or more occurrences. \(\)
	quoted parens	surround an arbitrary subexpression. \[0-9]
	quoted numbers	rematch a previous subexpression enclosed by \(\) \{\}
	quoted curlies	surround a repeat count for the preceding subexpression.

The caret (^) and dollar sign ($), which represent beginning and end of line respectively in the UNIX editors, do not have that meaning in JYACC FORMAKER regular expressions.

5.6 Field Edits

Field edits generally control the processing of data that have already been keyed into a field. Thus, right justified and upper case modify the appearance of data on the screen, while data required and must fill call validation routines at field exit. To modify field edits:

1. Position the cursor anywhere within the field.

2. Hit the PF4 key to bring up the field characteristics window (Figure 9).
3. Choose field edits to bring up the field edits window (Figure 13). The window will show any field edits that are currently set.
4. To turn on a field edit enter y; to turn it off, enter n or space. More than one field edit may be active.
5. Hit TRANSMIT to effect the change (or EXIT to abort).

At this point, the field characteristics window will still be displayed. You may then choose exit or hit the EXIT key to close the window, or set other field characteristics by choosing another option.

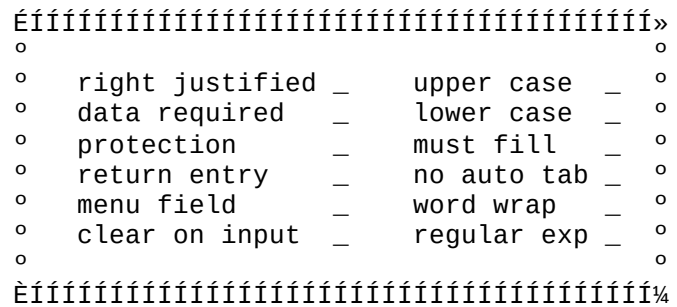

Figure 13: Field Edits

The following field edits are available. Unless otherwise stated, the actions described are performed by the library routine openkeybd.

5.6.1 Right-justified Fields

In a right justified field, characters are entered starting at the rightmost position. As each additional character arrives, previously entered text shifts one position to the left. When you tab into a right-justified field, the cursor goes to the right-hand edge.

If you activate insert mode using the INSERT key, characters entered in the middle of the field cause text between the cursor and the start of the field to be shifted left. The DELETE CHAR key shifts all characters to the left of the cursor one position to the right. The FIELD ERASE key clears the entire field, regardless of cursor position.

5.6.2 Data Required Fields

A data required field is checked at field exit for the presence of at least one non-blank character. If the field is digits-only, it must contain at least one digit; punctuation doesn't count. This validation is performed when the last position of the field is filled, or when you attempt to leave the field using the TAB, NEW LINE, or arrow keys. If the field is blank, JYACC FORMAKER displays an error message and resets the cursor to the beginning of the field (according to its justification).

5.6.3 Protected Fields

You can protect fields against the actions listed in figure 14, which pops up when you type y after protected in the field edits window. Protections may be combined arbitrarily; just type y after those you want to turn on. Use n or space to turn a protection off.

If a field is protected from data entry, all data characters typed there will be rejected, with a beep. Often it makes sense to protect such a field from tabbing into as well. One case where it does not is a circular scrolling field containing a list of choices, from which you select by scrolling to the

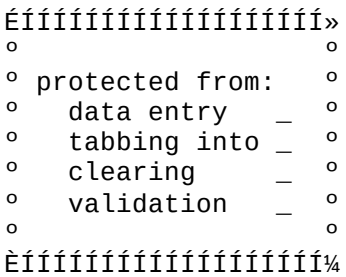

Figure 14: Field Protection Window

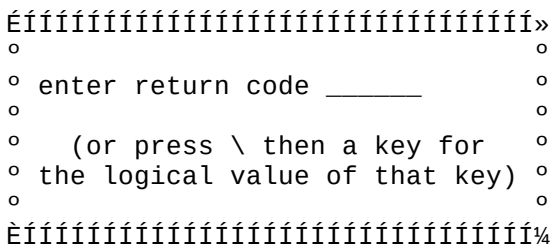

Figure 15: Return Code Window

appropriate one. Another is a shifting field, which a user must tab into in order to view the offscreen data.

If a field is protected from tabbing into, you will not normally be able to move the cursor there. The TAB, BACKTAB, arrow, and RETURN keys all cause the cursor to skip over tab-protected fields. Under certain circumstances, the arrow and HOME keys may place the cursor in a protected field; refer to the library functions ok_options and home, respectively.

If a field is protected against clearing, then the CLEAR ALL, ERASE, and DELETE CHAR keys, plus the clear-on-input field edit and certain library functions, will not erase its contents.

Finally, if a field is protected against validation, none of the normal field-exit checks will be performed, either when the field is tabbed from or when the entire screen is validated. Character edits, and certain field edits such as case and justification, will still be enforced.

In draw mode, all field protections are suspended.

5.6.4 Return-entry Fields

Note: return-entry fields are no longer used in menus. See the following section.

Normally, the library function openkeybd processes an entire screen, returning control to its caller only when the TRANSMIT key or another "returnable" key is struck. If your program needs to receive control at a certain field, however, you may designate that field as return entry. When such a field is filled or tabbed from, openkeybd validates the field, leaves the cursor there, and returns immediately to its caller.

When you type y after return entry on the field edits menu, the window in Figure 15 will pop up. In this window you may enter an integer code for openkeybd to return from the field; this makes it easy for your program to tell what field to process. You may enter the return code in any of several formats:

- a decimal number: 50
- an octal number, with leading 0: 062
- a hexadecimal number, with leading 0x: 0x32
- an ASCII character, with surrounding apostrophes: '2'
- a JYACC FORMAKER key mnemonic, as an alphanumeric string. You may also press the \ key, then the desired function key.

If you do not specify a return code in this window, openkeybd applies a set of default rules to determine its return value. When a field is return-entry, the application program must provide for positioning the cursor to the next field (perhaps by calling tab). If a field is return entry and scrollable, the application program must provide for scrolling, by calling rscroll. Refer to the library function openkeybd for more details.

5.6.5 Menu Fields

Note: menu fields perform the functions provided under Release 3 by return-entry fields in menus.

In menu screens processed by the library functions menu_proc and choice, and in item selection windows, only fields tagged with the menu field edit are eligible for selection. You can additionally choose the value to be returned by menu_proc, in the window that pops up when you enter y after menu field in the field edits window:

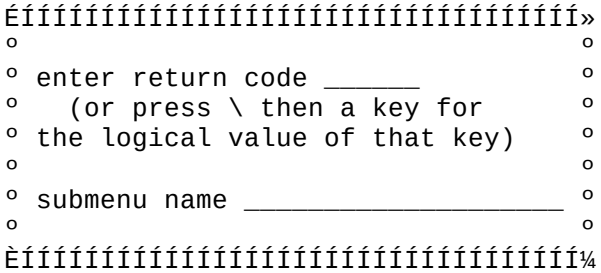

Figure 16: Menu Field Window

The rules for entering the return code are exactly as in Section 5.6.4. You may also enter the name of a sub-menu window, after submenu name, to achieve one level of pull-down menus; in this case, specifying a return code is useless, since the return value will come from the sub-menu.

Sub-menus work as follows. When the cursor enters a menu field with a sub-menu, the sub-menu window automatically pops up. All the cursor-motion keys operate as normally in the sub-menu, except the left and right arrows; they move to the next or previous selection in the main menu, and bring up a new sub-menu if it has one. Selecting an item from a sub-menu causes its value to be returned to the caller of menu_proc. Consequently, the main menu items themselves are never returned; your program only needs to worry about the return values of sub-menu items.

5.6.6 Clear-on-input Fields

For some kinds of fields, such as right-justified and currency fields, it is easier to type the field's contents afresh when a change is desired than to overwrite part or all of the field. Such fields may be designated clear-on-input; then, the first data character typed after the cursor enters the field causes it to clear, and that character stands alone in the field.

Right-justified and currency fields are not made clear-on-input by default.

5.6.7 Upper- and Lower-case Fields

These edits convert any alphabetic characters entered into the field to upper or lower case. This is valuable for consistency of appearance, and simplifies such tasks as database lookups. A field may be set to either upper case or lower case, but not both.

5.6.8 Must-fill Fields

A must-fill field is valid if it is empty, or if it contains no blanks whatsoever; leading, trailing, or embedded blanks with data characters are invalid.

5.6.9 No-autotab Fields

Normally, when you fill the last position of a field, the cursor will jump to the beginning of the next one. In a no-autotab field, however, the cursor will remain at the last position of the current field. Further input will either overwrite the last position or, if the OK_ENDCHAR option is set, be rejected with a beep (see ok_options). You must use the TAB, RETURN, or other cursor motion key to get to exit the field.

5.6.10 Word Wrap Fields

Word wrapping automatically transfers text from occurrence to occurrence of an array, to prevent "words" from being broken across lines. A "word" is simply a string of non-blank characters. Word wrap may be used only on fields that have been made scrollable (Section 5.3.3), or arrays (Section 5.3.2), or both. The fields may or may not be shiftable; they must be unfiltered (Section 5.5), and may not have a field regular expression. Within an array or field that uses word wrap, the following rules apply:

- Spaces separate words. You type text in freely; when a non-blank character occurs at the last position of a line, the entire word of which it is a part is removed from that line and inserted at the beginning of the next. If that line has insufficient room, the end of that line is also wrapped; wrapping propagates, if necessary, to the end of the array or scroll. If there is insufficient room on the last line, nothing moves, and the terminal beeps.
- Blank lines separate paragraphs. Whenever word wrap moves text into a blank line that is followed by additional lines of text, the blank line is preserved, and all following lines of text move down by one.
- If the DELETE CHAR key is hit and there is no text to the right of the cursor, as much text from the following line as will fit is moved to the cursor's current position; subsequent text moves up appropriately. Hitting DELETE CHAR at the beginning of an empty line deletes that entire line. DELETE LINE will delete a full line of text.
- The RETURN key positions the cursor to the next line of text within the current array or scroll, regardless of other fields on the screen, until the last (maximum) line is reached. It then positions the cursor to the first field past the current line.
- In insert mode only, the RETURN key opens up a line. Any text following the cursor on the same line is moved to the next line, and any additional lines of text are moved down by one. The INSERT LINE key does the same thing, whether insert mode is active or not.

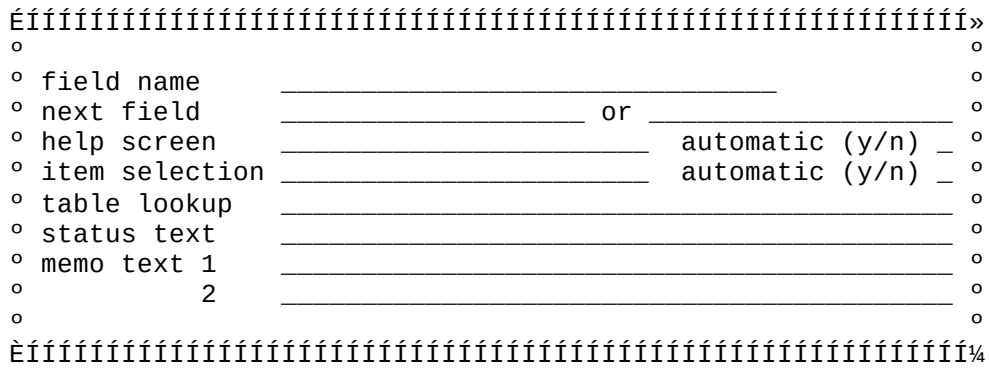

Figure 17: Field Attachments Window

5.6.11 Field Regular Expressions

You may attach a regular expression to a field, thus requiring that its contents match the regular expression when you exit the field. When you enter y after regular expression in the field edits window, a small window (Figure 12) pops up, prompting you to type in the expression there.

See Section 5.5.1 for a full explanation of regular expressions.

5.7 Field Attachments

Field attachments are chiefly names of other objects associated with a field, such as its help screen, although the field's name is itself an attachment. You can alter a field's attachments by bringing up the window in Figure 17, thus:

1. Bring up the field characteristics menu by hitting PF4.
2. Choose attachments. The field attachments window will be brought up, with current attachments (if any) displayed; see Figure 17.
3. Enter or erase the desired data, as decribed in subsequent sections.
4. Hit TRANSMIT to save the field attachments, or EXIT to abort.

5.7.1 Field Name

A field name entry assigns the field a name, to which an application program can then refer. The first character of the name must be alphabetic, and the rest must be either alphanumeric or underscores; no blanks are allowed. Upper and lower case letters are considered distinct, so that "field1" and "Field1" are different names.

Although field names are optional, they are strongly recommended. An application program can also refer to fields by number, which may be more convenient at times. However, names can be permanently assigned to fields, while field numbers change whenever a field is moved, inserted, or deleted from a screen (Section 5.1). Names are also more convenient if you plan to generate programming language data structures from your screens.

Only one field name can be assigned to an array (Section 5.3.2); individual array elements can be referenced by field name and element number.

When a field is copied (Section 5.2), the new field retains all the characteristics of the original except its name.

5.7.2 Next Field

A next field entry designates the field to be tabbed to when this field is exited. Normal tabbing is from left to right and top to bottom (the same order as field numbering), except that tab-protected fields are ignored. A next field entry is ignored if the target field is nonexistent or tab-protected; in such cases, the next-field designation is said to fail. The field attachments window provides for two next-field designations; if the first fails, the second is tried.

The next-field designation does not affect the BACKTAB or RETURN keys. It may affect the right-arrow key, if horizontal arrow behavior is set to OK_TAB using ok_options; other arrow keys are unaffected.

Next fields can be designated by name, or by absolute or relative number. Field numbers are assigned during screen compilation (Section 5.1); precede them by a # sign (#1, #14) for absolute, or a plus or minus (+1, -5) for relative. Use either +0 or -0 to designate the current field. Names are assigned using the field attachments window (preceding section), and should be entered without adornment. If you specify a next field by number, and later insert or delete a field, the resulting tab operation may be quite different from what you intended.

You can designate a particular array element or scrolling item by attaching a subscript, the occurrence number, to a field designation. The number may again be either absolute (no sign) or relative (plus or minus sign). It may be attached in two ways: enclosed in brackets (alpha[5], beta[+1], #5[1]), or preceded by a colon (alpha:5, beta:+1, #5:1). The colon form is obsolescent. Occurrences are explained in Section 5.3.

If a next-field edit belongs to an array or scrollable field, then whenever the operator tabs from an occurrence of the array or field, the same next-field designation is used. In this case, the current occurrence number is saved; if the designated next field is also either scrollable or part of an array, but the next-field edit contains no occurrence, the saved occurrence is tried. If it is greater than the number of occurrences in the destination field or array, the next-field option fails.

As an example of next-field use, suppose you have two parallel scrolling arrays named array1 and array2, and you wish the cursor to tab through them column-wise (the default is row-wise). You would designate the next fields as follows:

```
                    array1         array2

     primary        array1[+1]     array2[+1] alternate
                    array2[1]      array1[1]
```

These designations would move the cursor to the next occurrence of each array, until the last was reached; then the primary designation would fail, and the secondary would take the cursor to the first occurrence of the other array. If the arrays are scrolling, this will also cause the TAB key to scroll them.

5.7.3 Help Windows

The help screen option enables you to name a window to be displayed whenever the HELP key is hit while the cursor is within the field. If automatic help is specified, the help window will be displayed as soon as the field is tabbed into, if the field's contents have not been validated (i.e. its VALIDED bit is not set). The creation of help windows is described in Section 6.2. There are several types, including display-only text, menus for more detailed help, and help windows allowing data entry.

A help window can also be specified on a screen basis; see Section 3.6. You may specify both a help window and an item selection screen (Section 5.7.4) for a field, but they will conflict, with unpredictable effects.

The name of the help window may optionally be preceded by the location on the screen where it should appear. You write the line and column where the window's upper left-hand corner should go, enclosed in parentheses. The following example specifies a window named customer.hlp, to be placed at line 5 and column 10 of the screen:

```
help screen: (5,10)customer.hlp_____________
```

If you do not supply a location for the window, JYACC FORMAKER will automatically bring it up where it does not hide the field it is attached to.

5.7.4 Item Selection

The item selection option enables you to name an item selection window which will be displayed whenever the HELP key is hit while the cursor is within the field. An item selection window contains a list of valid field entries, from which the operator selects one; the selected item is then copied to the underlying field, and the window closed. Selection operates according to the rules defined at choice in the Programmer's Guide; basically menu rules. The creation of item selection screens is described in Section 6.3.

If automatic item selection is specified, the item selection window will be displayed whenever the field is tabbed into and the field has not been validated since it was last changed. Although you may specify both a help window (Section 5.7.3) and an item selection screen for a field, they will conflict and the effects will be unpredictable.

Item selection, even if automatic, does not prevent an operator from entering data into the field ad lib. You may want to use a table lookup screen (see below) to restrict a field to some list of entries; in fact, you can use a single screen for both item selection and table lookup.

The name of the item selection window may optionally be preceded by the location on the screen where it should appear. You write the line and column where the window's upper left-hand corner should go, enclosed in parentheses. The following example specifies a window named areacodes, to be placed at line 12 and column 55 of the screen:

```
item selection: (12,55)areacodes_______________
```

If you do not supply a location for the window, JYACC FORMAKER will automatically bring it up where it does not hide the field it is attached to.

5.7.5 Table Lookup

A table lookup window is very similar to an item selection screen, except that the list of items it contains is used to validate a field entry. The window is never actually displayed; the contents of the field being validated are compared with the items it contains, and the field fails validation if none matches. Windows used for table lookup validation are ordinarily also used for item selection. For further information see section 6.3.

5.7.6 Status Text

Status text is a message to be displayed on the screen's status line whenever the cursor is within its field. The message can be used as a prompt to the operator, explaining what entries are required or appropriate to that field. You can embed function key names and change display attributes within such a message; refer to the library function d_msg_line for details.

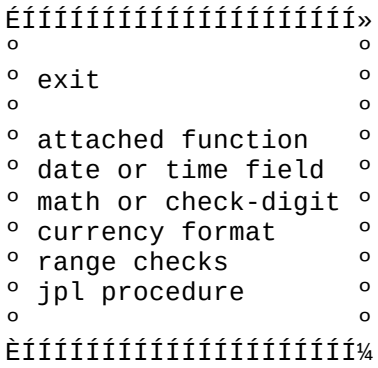

Figure 18: Miscellaneous Special Edits Menu

5.7.7 Memo Text

The memo text option enables you to mark fields for special processing by an application program. Nine lines of memo text are available, although only two are visible in the field attachments window at one time. You enter memo edits into a scrollable and shiftable array; you can get to the rest of the data with the arrows or the ZOOM key.

Any or all of these edits can store additional information about a field. Each edit is saved as a string, which can be retrieved using edit_ptr and examined by the application program. See the JYACC FORMAKER Programmer's Guide for examples.

Memo edits are not validated in any way.

5.8 Miscellaneous Special Edits

There are several more field characteristics that require relatively detailed information from you. To get at them,

1. Position the cursor anywhere within the field.
2. Hit the PF4 key to bring up the field characteristics window (Figure 9).
3. Choose misc. edits to bring up the menu in Figure 18. Initially, exit is displayed in reverse video. To select an edit, either:

    a. Using the TAB, BACKTAB, space, BACKSPACE, or arrow keys, position the reverse video area to the desired edit, and hit TRANSMIT, or
    b. Hit the initial letter of the desired edit.

4. In either case, a new window will appear, depending on the entry. You can exit from the menu by following the above procedure, or by hitting the EXIT key.

The field characteristics window will still be displayed; you may choose exit or hit the EXIT key to close the window, or set other field characteristics by choosing another option.

5.8.1 Attached Functions

An attached function names an application function associated with a field, to be called by the JYACC FORMAKER run-time system. A field may have two attached functions, which you enter in the window of Figure 19.

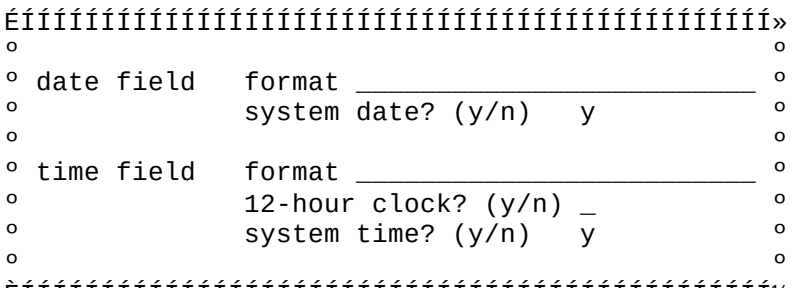

Figure 20: Date and Time Field Window

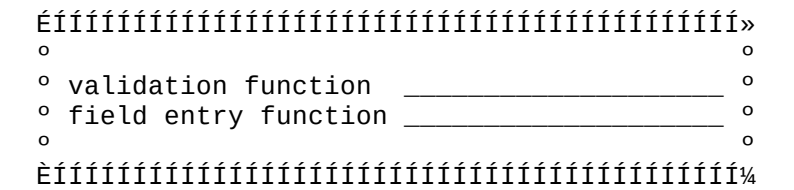

Figure 19: Attached Function Window

The validation function is called after tabbing or returning out of the field. It may be used to do any checking not supported by the other field edits available in JYACC FORMAKER, or to set up other fields that are conditional on the present one. The field entry function is called when the cursor enters the field. With care, it is possible to use the same function for both field entry and exit.

Attached functions written in a standard programming language must be registered in a function list in order to be called at run-time; refer to the section on attached functions in the JYACC FORMAKER Programmer's Guide. You can attach a procedure written in the JYACC Procedural Language (JPL) and stored in a file by typing jpl filename in the window; JPL procedure names need not be put into the function list.

5.8.2 Date and Time Fields

To create a date or time field:

1. Bring up the miscellaneous edits menu (Section 5.8).
2. Choose date or time field. The date and time field window will be brought up, with the current date or time edit (if any) displayed. See Figure 20.
3. Enter or erase the desired data.
4. Hit TRANSMIT to save the date or time data, or EXIT to abort.

A date field displays a date in a user supplied format, such as mm-dd-yyyy or DD/MM/YY. The format is not validated. It is used to format a date, by making the following substitutions. (Except where otherwise noted, either upper or lower case may be entered without affecting the result.)

- MM is replaced by a two-digit numeric month; ZM is replaced by a month of one or two digits, with leading zero suppressed.
- DD is replaced by a two-digit numeric day of the month; ZD is replaced by a day of one or two digits, with leading zero suppressed.
- DDD is replaced by a three-digit numeric day of the year, e.g. 33 for Groundhog Day.

- YY is replaced by the last two digits of the year; YYYY is replaced by the entire year.
- MMM is replaced by a 3-character alphabetic abbreviation of the month name. The case of each M determines the case of the corresponding letter in the name. In July, for example, MMM would be replaced by JUL, and Mmm would be replaced by Jul.
- DOW is replaced by a 3-character alphabetic abbreviation of the day of the week. The case of each letter in the format string determines the case of the corresponding letter in the day's name. On Wednesday, for example, dow would be replaced by wed, and dOw would be replaced by wEd.

All other characters are put literally into the formatted date. The above items may be supplied in any order, and any of them may be omitted or repeated within the field.

If system date is specified, the date initially displayed is the current date obtained from the operating system when the screen is brought up on the display. If the field is not protected from clearing, you can refresh the date by hitting the ERASE or CLEAR ALL key. Alternatively, you may enter a date which is then validated by the utility as to both content and format. If no system date is specified, the field is not initialized with the current date, but any date you enter will be validated as to both content and format.

A time field displays a time of day in a user-supplied format, such as hh.mm.ss or HH:MM a.m. The format string is not validated. It is used to format the time, by making the following substitutions. (Either upper or lower case may be entered without affecting the result.)

- HH is replaced by a two-digit hour; ZH is replaced by the hour, with leading zero suppressed.
- MM is replaced by a two-digit number of minutes; ZM is replaced by the minutes, with leading zero stripped off.
- SS is replaced by a two-digit number of seconds; ZS is replaced by the seconds, with leading zero suppressed.
- If the time is after noon, AM or A.M. is replaced by PM or P.M. If the time is before noon, PM or P.M. is replaced by AM or A.M.

All other characters are put literally into the formatted time. The above items may be supplied in any order, and any of them may be omitted or repeated within the field.

The time is given according to a 24 hour clock; if the format includes AM, A.M., PM, or P.M., they are treated as constant text and displayed as is.

If system time is specified, the time initially displayed is the current time obtained from the operating system when the screen is brought up on the screen. If the time field is not protected from clearing, you can refresh the time by hitting either the ERASE or the CLEAR ALL key. Alternatively, you may enter a time which is then validated by the utility as to both content and format. If no system time is specified, the field is not initialized with the current time, but any time you enter will be validated as to both content and format.

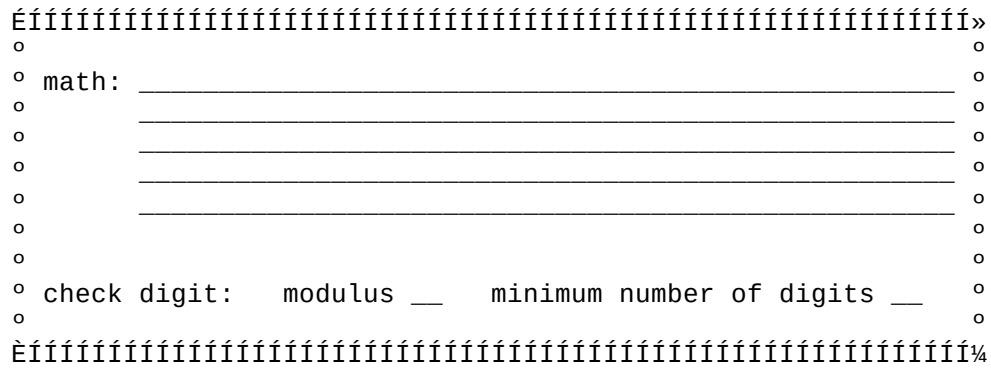

Figure 21: Math and Check-Digit Window

5.8.3 Math and Check Digit

To create a math or check digit edit:

1. Bring up the miscellaneous edits menu (Section 5.8).
2. Choose math or check-digit. The math and check-digit window will be brought up, with the currently set math or check digit edit, if any, displayed; see Figure 21.
3. Enter or erase the desired data.
4. Hit TRANSMIT to save the edit (or EXIT to abort).

The check digit option causes the field to be validated according to a standard check digit algorithm. Two algorithms, mod-10 and mod-11, are automatically supported, but others can be added. (The check digit routine, ckdigit, is included in the JYACC FORMAKER library. A detailed description can be found in the source code, which is included with the package.) Non-numeric characters in the field are ignored.

The math option causes JYACC FORMAKER to evaluate one or more user-defined calculations and place the results in fields. The calculations are performed when you fill or tab out of the field to which the expressions are attached, or hit the TRANSMIT key upon completing the screen. The calculated values are stored in whichever field you designate; the fields to which the expressions are attached need not appear in the expression. Multiple expressions must be separated by semicolons, even if they are on separate lines in the window.

A math expression start with an optional floating point size specification for the destination field. This specification has the form %m.n where m specifies the total number of characters in the output and n the number of digits after the decimal point. If no size is supplied, the total length defaults to the length of the destination field. The number of decimal places defaults to that given in a float or double data type edit attached to the destination field (see Section 5.9); if there is none, to the number of decimal places in an amount edit attached to the destination field (see Section 5.8.4); or to 2 if there is neither.

The optional size specification is followed by the destination field designation, an equal sign, and the body of the expression. The expression body can contain numeric constants, field designations, parentheses, and the arithmetic operations + - * / and ^ (raise to a power).

Fields and occurrences in math expressions may be designated by name or by absolute or relative number (preceded by the sign #), with an optional occurrence number. If the calculation is attached to an array or scrolling field, it is performed every time you fill or tab out of an occurrence of the array or field. In this case, the current occurrence number is saved as a default. If any field specified in the math expression is either scrollable or

part of an array but no associated occurrence number is supplied, the default number is used, if possible. If the default number is greater than the number of occurrences in the specified field or array, an error results.

Typical math expressions look like this:

```
%8.0 #3 = #1 * 12 + #2
fielda:2 = (fielda:1 - 6.235) / fieldb:1
```

In math expressions, the designation for the "current field" is #+0; the designations for the fields preceding and following the "current field" are, respectively, #-n (= n fields before the current field) and #+n (= n fields after the current field). As an example of this notation, consider the following two math expressions:

```
#-3 = #+3 * #+0

#-3 = #+0+6
```

In the first math expression, the field that occurs three fields before the current field is set to the value obtained by multiplying the current field's value by the value of the third field after the current field. Thus, if #+0 = 10 and #+3 = 5, then #-3 = 50. In the second math expression above, the value of the field three fields before the current field is equal to the sum of the current field's value and 6 (e.g., #-3 = 16, if #+0 = 10).

There are three special functions: @sum, @date, and @abort. @sum yields the sum of all occurrences in an array or scroll:

```
@sum array1
@sum #2
```

@abort followed by a number in parentheses passes the number to the library function isabort, q.v. That function causes the JYACC FORMAKER library to return control to the application as quickly as possible:

```
@abort(1)
```

@date yields the number of days elapsed between 1/1/1753 and the date given in following field item. The following field item may be a field name or number, or a literal date enclosed in parentheses. Literal dates must have the format MM/DD/YYYY. For instance:

```
@date quarterday
@date #-1
@date (3/31/1985)
```

An error results if the field designated by @date is not a date field, or does not contain month, day and year somewhere in its format. If the destination field is a date field, the number resulting from the calculation is interpreted as a number of days elapsed since 1/1/1753, and the resulting date is displayed according to the date field format. If field1 and field2 are both date fields,

```
field2 = @date field1 + 30
```

will set the date in field2 to 30 days past the date in field1.

5.8.4 Currency Formatting

A currency or amount field is formatted specially after you tab out. To create or change a currency format:

1. Bring up the miscellaneous edits menu (Section 5.8).

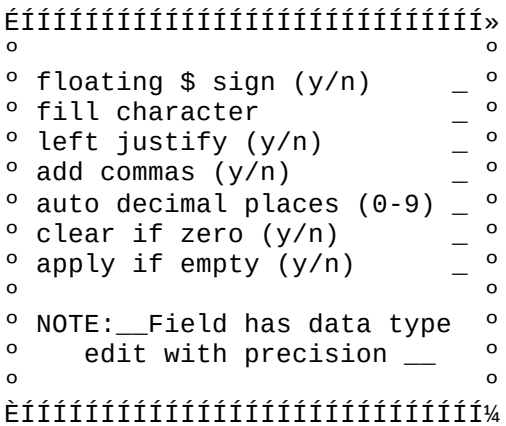

Figure 22: Amount Edits Window

2. Choose currency format. The window of Figure 22 will be brought up, with the current amount edit, if any, displayed.
3. Enter or erase the desired data. To get rid of a currency format altogether, use the CLEAR ALL key, or erase each field individually.
4. Hit TRANSMIT to save the edits, or EXIT to abort.

The following fields after the current field format parameters are supported:

floating $ sign	Enter y to insert a floating dollar sign, n or space to omit it. (For a fixed dollar sign, omit this option; enter a $ as display data on the screen, in the last character position preceding the field.)
fill character	Enter a fill character, if desired, or leave blank. The character specified will replace any blanks in the field.
left justify	Enter y for left justification, n or blank for right justification.

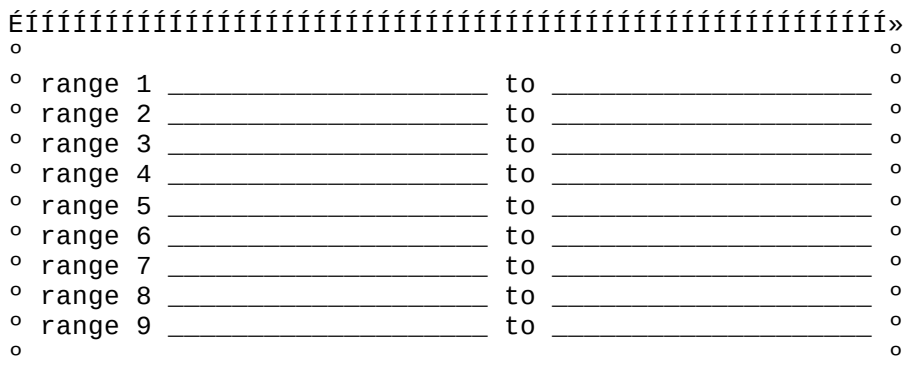

Figure 23: Range Window

add commas	Enter y to insert commas, n or blank to omit them.
auto decimal places	Enter the minimum number of decimal places to be displayed. More decimal places will be displayed if keyed in during data entry.
clear if zero	Enter y if the field should be cleared to blanks when the amount is zero, n or blank for a formatted zero amount, such as *******0 or 0.00.
apply if empty	Enter y if the amount edit should be applied to empty fields, n or blank if the edit should be skipped.

A 15-character-wide field with the following field currency format parameters:

```
floating $
fill with "*"
right-justified
add commas
two decimal places
```

will produce the following field transformations:

Plain	Formatted
123456789	$123,456,789.00
12.345	*********$12.35

5.8.5 Range Checks

To create or change range edits:

1. Bring up the miscellaneous edits menu (Section 5.8).
2. Choose range checks. The range window will be brought up with the current ranges, if any, displayed (Figure 23).
3. Enter or erase the desired data.
4. Hit TRANSMIT to save the ranges, or EXIT to abort.

The range option enables you to enforce minimum and maximum values for a field. If the field has been defined as digits-only, the values entered will be treated as integers. If the field is numeric, the values will be interpreted as floating point numbers. In all other cases, the values will be compared as character strings.

Range endpoints are restricted, both in length and character filter, to valid field entries. A field may have up to nine permissible ranges; you may omit either the lower or the upper limit for any of them. An empty field is

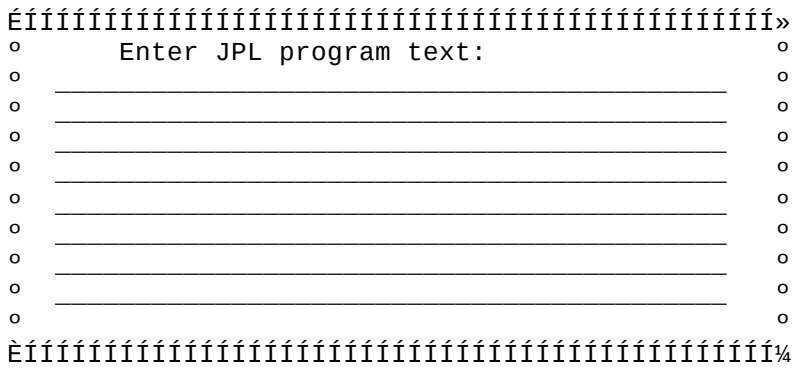

Figure 24: Attached JPL Procedure Window

considered in range; if data entered are out of range, the run-time system displays an error message on the status line.

5.8.6 JPL Procedure

The JYACC Procedural Language, or JPL, is an interpreted programming language designed especially for use in JYACC FORMAKER screens. You can attach JPL statements directly to a field, using the window in Figure 24

To create or alter an attached JPL procedure:

1. Bring up the miscellaneous edits menu (Section 5.8).
2. Choose jpl procedure. The window of Figure 24 will be brought up, with the current JPL code, if any, displayed.
3. Enter or erase the desired data. The window contains a scrolling and shifting array, in which you type your JPL program.
4. Hit TRANSMIT to save the changes, or EXIT to abort.

The attached JPL code will be executed during field exit processing, directly after the field validation function. Refer to the JPL Programmer's Guide for how to write JPL programs. As a special case, you can execute a JPL procedure stored in a file by typing jpl filename in this window.

5.9 Data Types

JYACC FORMAKER includes a utility, f2struct, that can generate a prgramming language data structure corresponding to the fields of a screen; see the Configuration Guide for usage. You can use the screen editor to assign data types to fields which that utility understands. To supply a field with a data type, or to exclude it from the structure:

1. Position the cursor anywhere within the field.
2. Hit the PF4 key to bring up the field characteristics window (Figure 9).
3. Choose type to bring up the data type window (Figure 25). Note that the labels in this window are taken from the message file, where you may adapt them to suit your primary programming language; the figure uses C language labels, as distributed by JYACC.
4. To set or change the option, either:

    a. Using the TAB, BACKTAB, space, BACKSPACE, or arrow keys, position the reverse video area to the desired option, and hit TRANSMIT, or
    b. Hit the initial letter of the desired option (o for omit, etc.). Either choosing exit or hitting the EXIT key will close the window without changing the type.

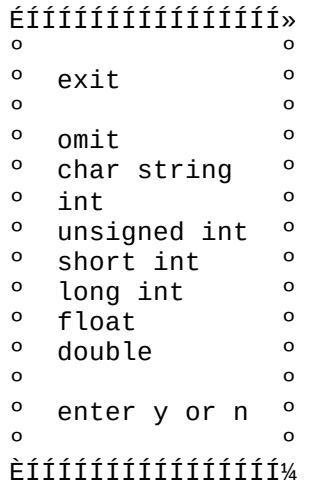

Figure 25: Data Type Menu

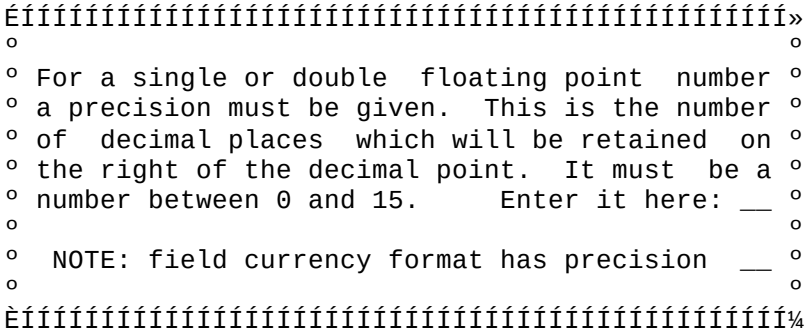

Figure 26: Precision Window

If you select a real data type (float or double in the figure), the precision window (Figure 26) will pop up. The default precision is the number of decimal places in the currency edit, if any, or 2 otherwise. To change the precision, enter the desired number and hit TRANSMIT. Hitting the EXIT key will leave the old value in effect.

5.10 Field Summary Window

xform provides a field summary window, a convenient way to inspect and change certain frequently modified field characteristics without wading through the PF4 menus. The window appears in Figure 5.10; to use it, you

1. Position the cursor anywhere in the field.
2. Press the PF5 key.
3. Enter any changes in the window. Only items in the first two lines may be modified; the rest are display-only, and are formatted similarly to a screen listing.
4. Hit the TRANSMIT key to commit your changes, or EXIT to discard them.

While the field summary window is open, you may press the PF4 key at any time, and alter field characteristics from its menu in the usual way. However, if you subsequently hit the EXIT key in the summary window, those changes will be lost.

In the remainder of this section we describe quickly how to use the window. The field characteristics themselves are fully described elsewhere in this chapter, and a reference is provided for each.

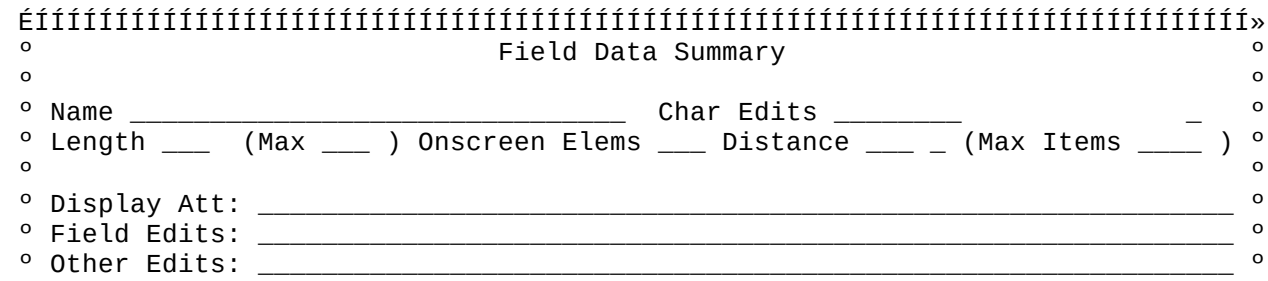

Figure 27: Field Summary Window

Name	The field's name, if any, is displayed here, and you may simply type in changes. (Section 5.7.1)
Char Edits	This is a circularly scrolling field (Section 5.3.3) displaying the field's character filter; use the up- and down-arrow keys to select the one you want. The one that is visible when you hit TRANSMIT on the window will be used. (Section 5.5)
Length	The field's onscreen length; type in any changes. (Section 5.3)
(Max)	The field's maximum shifting length. Modify directly. (Section 5.3)
Onscreen Elems	The number of onscreen array elements. Modify directly. (Section 5.3)
Distance	The distance from one array element to the next. Modify directly. (Section 5.3)
(Max Items)	The maximum number of scrolling items. Modify directly. (Section 5.3)
Display Att	All the field's display attributes. Not modifiable; refer to the lstform utility if you need an explanation of the attribute mnemonics. (Section 5.4)
Field Edits	Mnemonics for the field edits. Not modifiable; see lstform. (Section 5.6)
Other Edits	Mnemonics for the field's attachments and other special edits. Not modifiable, and the values of the edits are not given. (Sections 5.7, 5.8)

6 Special-Purpose Screens

The JYACC FORMAKER library includes routines for menu processing and automatic display of help screens. The following sections explain how to create forms for use with those functions. The functions themselves are described in the JYACC FORMAKER Programmer's Guide.

6.1 Menus

The library functions menu_proc and choice allow for easy selection of menu items. The current (or tentative) selection is always displayed in reverse video. To create a screen that can make use of this feature:

1. Create a selection field for each menu entry. Each field must be long enough to contain all the text to be displayed. You can make every field equal in length to the longest text to be entered in any of them, as on xform's field characteristics menu (Figures 28 and 29), or you can make them different lengths, as on xform's exit options menu (see Figure 2). The area defined as a field, not just the text, will be displayed in reverse video when the cursor is at an item.

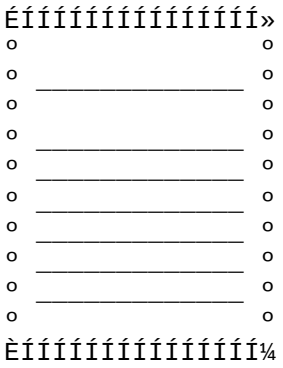

Figure 28: Fields Defined for Field Characteristics Menu

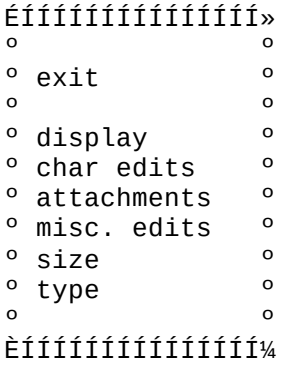

Figure 29: Initial Data Entered in Field Characteristics Menu

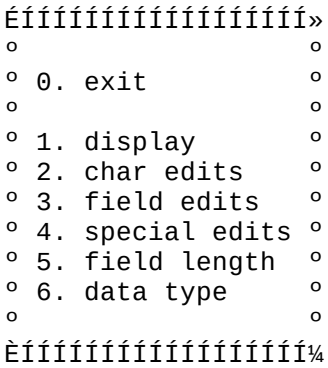

Figure 30: Menu Using Digits for Item Tags

2. Set the fields' display attributes as desired, but do not include reverse video.
3. Give each menu field a menu field edit (Section 5.6).
4. Enter the menu item text in the menu fields. Each item should start with a distinct character (Figure 29). The menu shown is used by xform for the selection of field characteristics (Figure 9); on the actual menu the fields are not underlined.

If possible, initial letters should be all upper case or all lower case. This permits the menu processing function to ignore the case of letters entered from the keyboard, treating a and A as equivalent. If several items would naturally start with the same letter, the item names can be preceded by digits (Figure 30), or by letters that are not part of the names themselves.

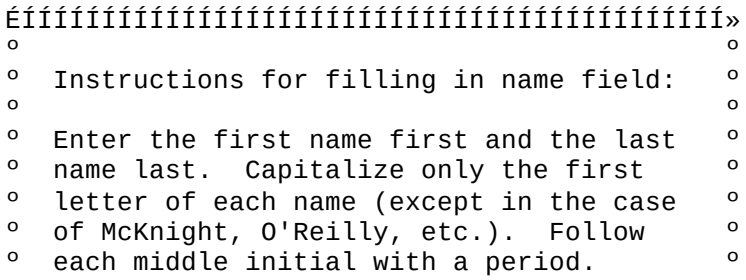

Figure 31: Help Screen with Display Data Only

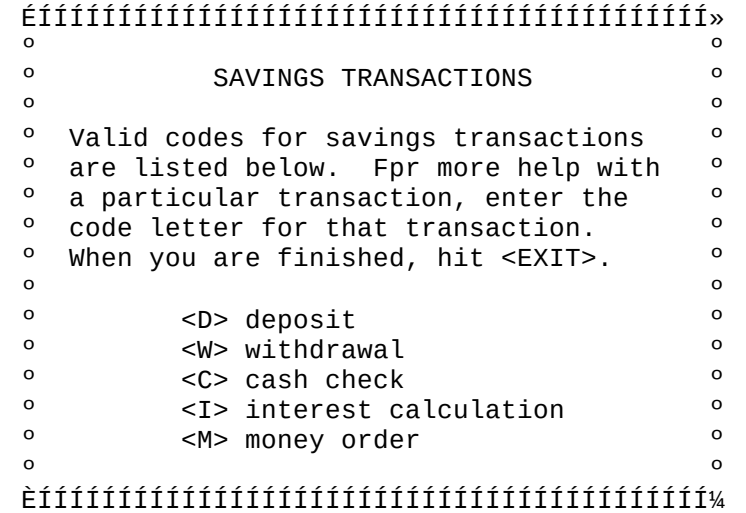

Figure 32: Help Screen Menu

Alternatively, upper and lower case initial letters can be mixed, allowing a distinction between a and A, but requiring the operator to use the shift key for upper case.

6.2 Creating Help Screens

JYACC FORMAKER's help screen feature associates a help window with a field or an entire form; each help screen is itself created with the JYACC FORMAKER authoring utility. Help screens can function in several different ways.

The simplest help screens, like Figure 31, are display-only; they contain no fields, just display data. When the HELP key is struck with the cursor in the associated field, the help screen is displayed as a window. The help window is closed at the next keystroke.

6.2.1 Help Screens Containing Menus

Help screens can contain menus which call up additional help screens. To create a help screen with a menu:

1. Create a screen with menu fields, as described in Section 6.1.
2. For each menu field, use the help screen option (Section 5.7.3) to specify an associated lower level help screen.

When the operator makes a selection from the menu, the associated lower level help screen will appear. The operator exits from the help screen menu by hitting the EXIT key.

6.2.2 Help Screens with Data Entry Fields

In some cases, it may be desirable for the operator to be able to enter data into a form while a help screen is displayed. This can be done if the help screen is created with a field for data entry.

When the help screen is displayed, the contents of the associated field are copied into the field on the help screen. You may then enter data in the help screen field. When you hit TRANSMIT, the data in the help screen field are copied back into the associated field, and the help window closes. If you hit EXIT instead, the field is not copied.

The help function will provide automatically for data entry whenever the help screen contains exactly one unprotected field that has no associated procedure name. The data entry field is normally defined to be as long as the field(s) the help screen is associated with. If the data entry field is too short for the data, the help function will automatically make the field shiftable, with a maximum length equal to the associated field length. The help function will also copy from the associated field:

1. the character edits (Section 5.5)
2. the following field edits (Section 5.6):

   - right justified
   - upper or lower case
   - data required
   - must fill

3. the following special edits:

   - check digit (Section 5.8.3)
   - currency format (Section 5.8.4)
   - range (Section 5.8.5)

The help function will not process both data entry and menu fields on the same help screen. However, it is possible to provide additional help screens for a data entry help screen, by using protected fields, as described in the next section; see Figure 33.

6.2.3 Help Sub-screens Using Protected Fields

An alternative to menu processing uses fields containing explanatory text, which are protected from data entry but not from tabbing into. When the cursor is positioned within such a keyword field and the HELP key is hit, the associated lower level help screen is displayed. To create a help screen using this feature:

1. Create a help screen with all text entered as display data.
2. Replace the key words by underscores, and hit TRANSMIT to compile the underscores into fields.
3. Re-enter the key words in the fields. Make the key words easily distinguishable from the rest of the text by use of distinctive display attributes, or by some other means such as capitalization.

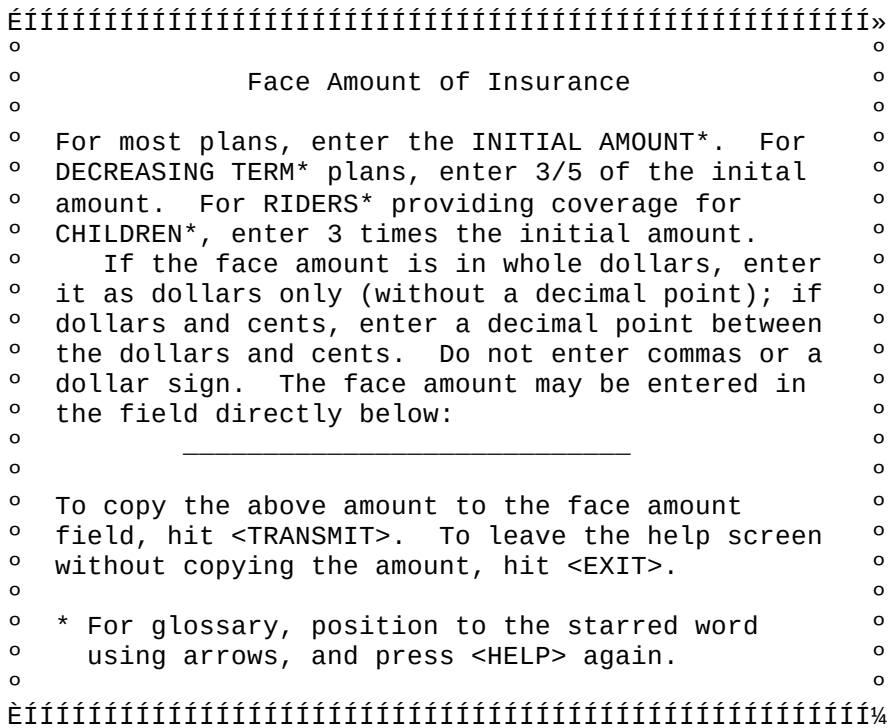

Figure 33: Help Screen with Data Entry Field and Protected Fields

4. Make each key word field protected from data entry, but not from tabbing (Section 5.6), and give it a help screen (Section 5.7.3).

The help function will not process both menu fields and protected keyword fields on the same help screen.

6.3 Item Selection and Table Lookup Screens

An item selection screen is similar to a help screen attached to a field. The contents of each field on the item selection screen correspond to a valid entry in the associated field; you choose an item by a process similar to menu selection. Because a list of valid entries for a given field can run to more items than could be displayed on the screen at one time, item selection screens may need to use scrollable arrays.

If there are many items, several could easily start with the same character, so the "initial character" selection feature is modified. Entering the initial character of one of the items simply positions the cursor to the next item on the list that starts with that character. (Note that, if the fields are defined as a scrollable array with offscreen data, this next item might not be visible at the time the character is keyed in.) The TRANSMIT key must be hit to complete the selection.

When you hit the TRANSMIT key, the item selection window is closed, and the selected item is copied into the associated field. If the EXIT key is hit, the window is closed but nothing is copied.

An item selection screen should consist of several fields, a simple array, or a scrollable array; all must be menu fields. If a scrollable array is used, the actual number of items entered should be the same as the maximum number. The lengths of fields on the item selection screen need not be the same as the length of the associated field. If the item selection fields are longer, as in Figure 34, only the first part of the field is copied, but the rest of the field can contain other information that is helpful to the operator.

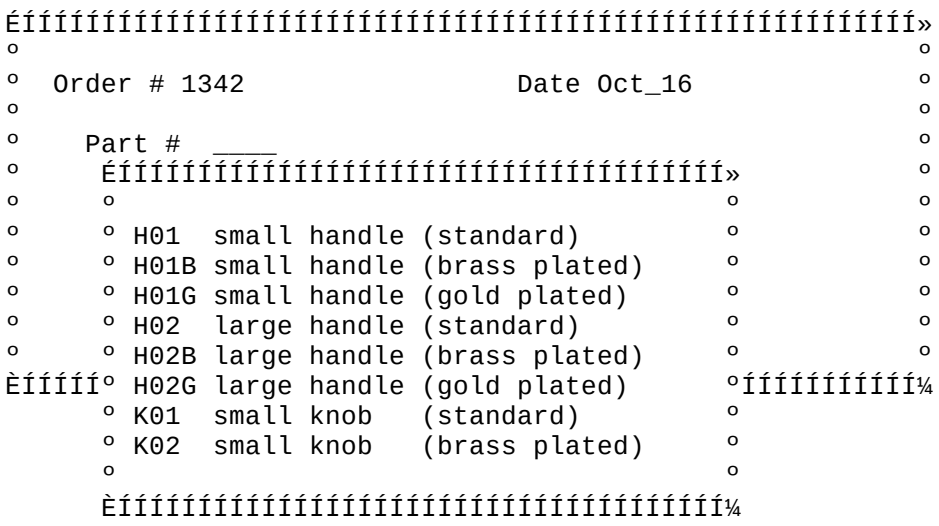

Figure 34: Item Selection Screen

An item selection screen is frequently also used as a table lookup screen, to ensure that a field's contents actually match one of the items; or it may be used for lookup alone. Lookup screens are created in exactly the same way as item selection screens; however, they are never displayed at run-time, so no attention need be paid to the appearance of the screen.

Index

In this Index, library functions are displayed in boldface, without the prefixes specific to the language interface. Video and setup file entries appear in ELITE CAPS, while utility programs and JPL commands are in elite lower-case. Function key names are in ROMAN CAPS.