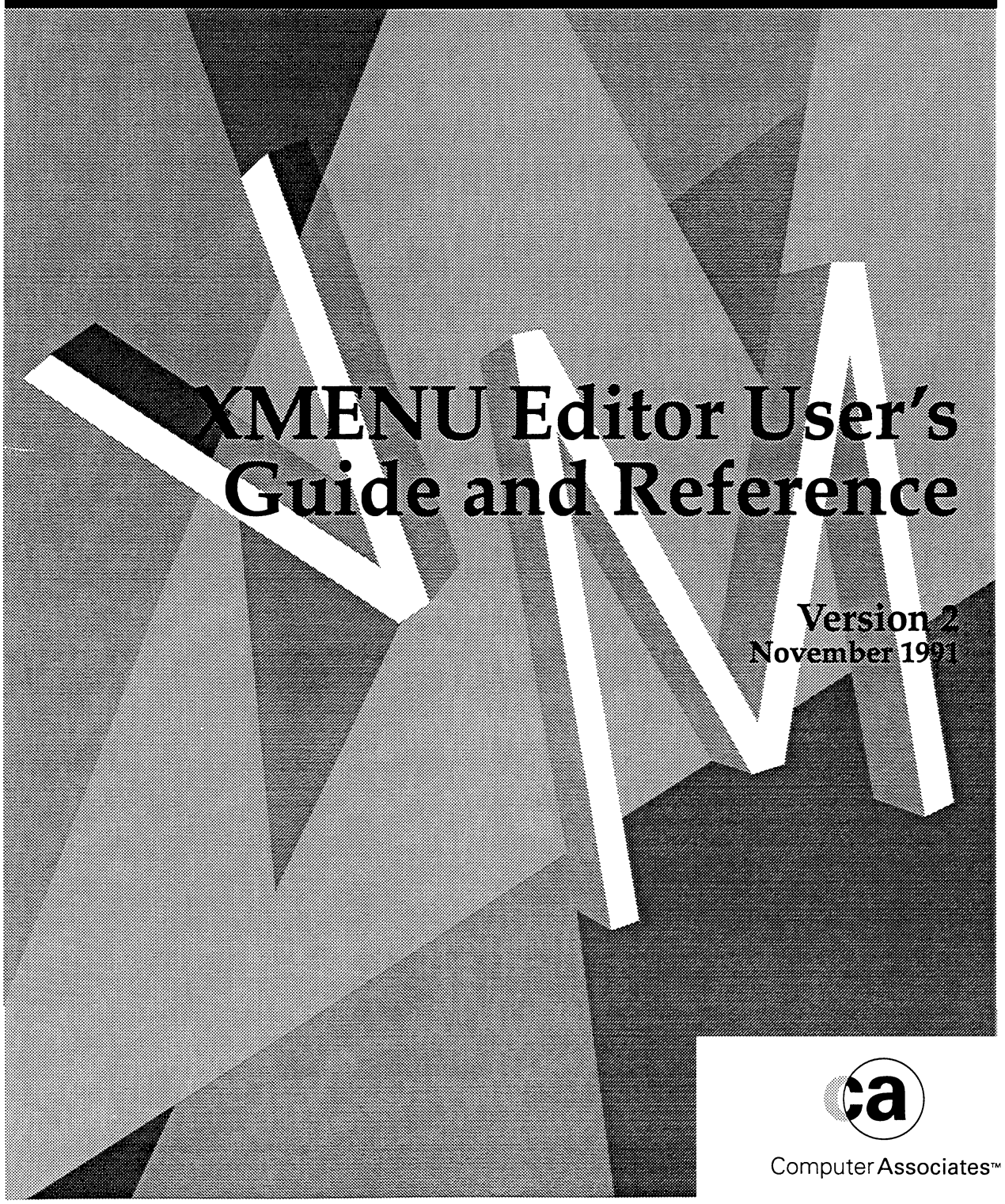
XMENU



XMENU Editor User's Guide and Reference

Version 2
November 1991

Computer Associates™

030510510301

Preface

The *XMENU Editor User's Guide and Reference* contains information for use with the Relay Technology product XMENU.

Audience

This manual is intended both for end users who want to create menus for applications that run in the VM environment, and application programmers who design and implement 327*x* full-screen applications. The *XMENU Editor User's Guide and Reference* describes how to create a menu, how to change the layout or display of a menu, how to define input and display fields on a menu, and so on.

The *XMENU Editor User's Guide and Reference* also includes a complete reference section for the XMEDIT command and for other XMENU utilities you can use with the product editor.

A basic understanding of XEDIT is assumed for anyone using this manual to design menus. If you have never worked with XMENU, a good place to start is with *XMENU in Minutes: An end-user's guide to creating menus.* This quick tutorial will give you some helpful, basic information on and experience with the XMENU editor.

How this manual is organized

This manual contains the following parts, chapters, and appendixes:

- Part 1, The XMENU editor user's guide

  Chapter 1, "Introduction to XMEDIT, the XMENU menu editor" on page 3, provides a general overview of the XMENU editor and includes examples of menus you can create using it.

  Chapter 2, "Creating menus with XMEDIT" on page 7, begins with a brief summary of the steps that are usually involved in creating a menu, and then takes you step-by-step through that process.

- Part 2, The XMENU editor utilities reference

  Chapter 3, "XMEDIT" on page 55, details the command syntax, options, subcommands, messages and return codes for XMEDIT, the XMENU menu editor.

  Chapter 4, "XMENULIB" on page 107, explains the use of and command format for the XMENULIB utility.

  Chapter 5, "PSEDIT" on page 109, describes the use of and command format for the PSEDIT utility.

  Chapter 6, "XMENUCOB" on page 113, is the command reference section for the XMENUCOB utility, and includes its command syntax, usage notes, and messages and codes.

  Chapter 7, "XMENUPLI" on page 115, provides the command syntax, usage notes, and messages and codes for the XMENUPLI utility.

Appendix A, "Some notes on menu design" on page 117, briefly discusses menu design considerations and lists references on the subject.

Appendix B, "XMEDIT output files" on page 119, describes the five output files XMEDIT can create.

Appendix C, "Attribute types and values" on page 121, provides a complete listing of field attributes you can assign with XMEDIT.

Appendix D, "Assigning attributes using the OLDWAY full-screen menus" on page 123, describes an alternate, "old way" of assigning attributes in Input mode that uses a full-screen menu instead of a selection bar with pull-down menus.

Appendix E, "Naming fields by using the OLDWAY full-screen menus" on page 125, describes an alternate, "old way" of naming fields in Input mode that uses a full-screen menu instead of a pop-up window.

Appendix F, "Changing the defaults by editing PROFILE XMEDIT" on page 127, describes how to tailor the default menu creation environment to meet special needs.

Appendix G, "Some useful XMEDIT Macros" on page 131 documents some useful XMEDIT macros to help generate and inspect fields on a menu.

Other manuals you should have

In addition to this manual, you should also have the following XMENU documents:

- *XMENU in Minutes: An end-user's guide to creating menus*
- *XMENU Subroutine Library Reference*
- *XMENU Utilities Reference*

Contents

Part 1. The XMENU editor user's guide

This section provides a complete guide for using XMEDIT, the menu editor for XMENU. Here's a "road map" of the major topics discussed in this section:

Chapter 1. Introduction to XMEDIT, the XMENU menu editor

The XMENU menu generator/editor is called XMEDIT. This is the tool you use to create and modify menus for full-screen applications.

XMEDIT provides you a "What You See is What You Get" (WYSIWYG) environment for creating menus: what you see on the screen during the creation and editing process is what you get when your application displays the menu.

XMEDIT presents you with a blank screen on which you place text for the menu title, field captions, and other information you want to display. When the text is in position, you use PF keys to specify how the fields will be presented to the user, and to define the fields as ones that will either accept input from the users or simply display information to them.

The three examples that follow illustrate a few of the types of menus that you can create using XMENU.

The example shown in Figure 1 is a true menu, where the user selects a transaction from a number of choices. It is also possible to enter CP or CMS commands from this menu.

```
                        PERSONNEL DATA MENU

                         Transaction Type


                    1.   View/Update Personnel Record
                    2.   Add New Hire
                    3.   File Disability Claim
                    4.   Track Resume
                    5.   View Job Description
                    6.   View Organization Chart


                    9.   Exit to CMS

Enter the number of the transaction you want to perform.

Selection===>

COMMAND==>
```

Figure 1. Sample menu screen

Figure 2 illustrates a screen for data entry. This screen contains prompts for field entries, space for the user to type entries, and some extra information to help the user.

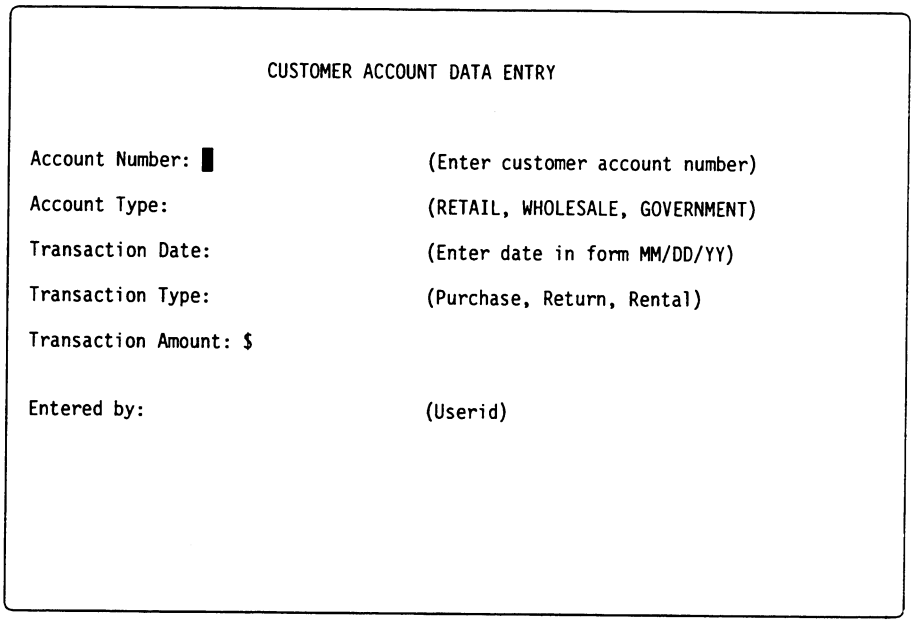


Figure 2. Sample data entry screen

Figure 3 shows a type of query screen. The user fills in known information, such as an employee number; the same screen might be used by the program to display the rest of the information.

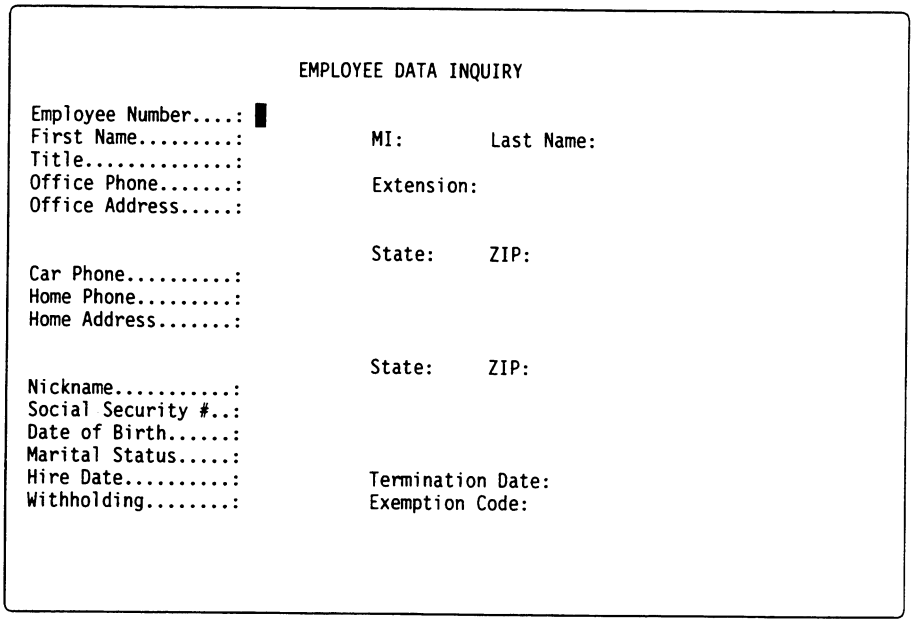


Figure 3. Sample query screen

No matter what words you use to describe XMENU-generated menus—panels, displays, menus, or screens—the important point to remember is that XMENU menus are not restricted to the multiple-choice format; they can also be used for data entry, data display, or combinations of the above.

Designing clear and useful menus requires careful thought about the users of your application and what they will be trying to accomplish. Appendix A, "Some notes on menu design" on page 117, discusses menu design and lists some references on the subject.

In addition to providing an interactive menu creation environment, the XMENU menu editor comes with several additional utilities:

XMENULIB Allows the collection of XMENU menus into CMS library files.

PSEDIT Edits 327*x* programmed symbol sets, such as are used when displaying Hebrew or Katakana characters, for example.

XMENUCOB Creates a COBOL structure for use by the MDXSCR subroutine.

XMENUPLI Creates a PL/I structure for use by the MDXSCR subroutine.

XMEDIT is described in detail in Part 1, "The XMENU editor user's guide" on page 1 and included in Part 2, "The XMENU editor utilities reference" on page 53, where the other utilities listed above are also detailed.

Chapter 2. Creating menus with XMEDIT

This chapter begins with an overview of how to create menus using XMEDIT and then takes you through each step of the menu creation process.

General procedures for creating a menu

To create a menu or screen using XMENU, the same general procedures always apply:

Steps for creating a menu

- Enter XMEDIT and specify the options that apply to your entire menu, such as the menu name and screen size.
- In Input mode, type the field captions and other text where you want them to appear on the screen.
- Transfer to Edit mode to rearrange these elements as necessary.
- Return to Input mode to define the input and display fields and give them display attributes.
- Set the position where the cursor should appear when the menu is first displayed.
- Name the fields that will receive or display data.
- Transfer to Display mode to test the menu by typing data in the fields.
- Print the menu if desired.
- Save the menu.

As you become accustomed to XMEDIT, you can go through some of these steps in a different order, or combine some steps to suit your work style. You can also stop at any point in the procedure and exit from XMEDIT, to finish your menu at another time, or to start over again.

To enable you to carry out these steps, XMEDIT functions in three modes:

- **Input mode**
- **Edit mode**
- **Display mode**

You use terminal keys (the ENTER key, Program Function (PF) keys, and Program Attention (PA) keys) in XMEDIT to move from one mode to another or to exit, and to perform a variety of tasks. Keys perform different tasks, depending upon which mode you are using. You can always press PF3 or PF15 to return to the previous screen or mode; ultimately, PF3/PF15 exits XMEDIT, saving your work.

The remainder of this chapter describes in detail the steps for creating and modifying a menu in XMEDIT. Each step begins with the basic instructions, an illustration of the screen, and a description of how the PF keys work for you. Each step also contains a section with notes on the details; these may not be of interest to you during the first time you use XMENU, so you can move ahead whenever you want to.

Entering XMEDIT and selecting options for your menu

To enter XMEDIT, type the following command at the CMS prompt and press ENTER:

```
XMEDIT
```

The command displays the XMENU Menu Generator/Editor (XMEDIT) Options Screen, Figure 4.

```
---------- XMENU 2  XMEDIT 3.00 Menu Generator/Editor ----------
            * * * Copyright 1981,1989 Relay Technology * * *
 Menu Filename            Library name            - The menu to be edited.
USING Filename            Library name            - Menu to use as prototype.
Use Profile  PROFILE  - Profile macro executed at start; clear if NOPROF wanted

Menu Characteristics: (Y|N - except for SIZE)       43  /  80  Size in rows/cols
 N  327x extended attributes          N  Set MDT for unprotected fields
 N  Set Skip for protected fields     N  Sound alarm on display

Work Setting: (Y|N)
 N  Use old XMENU menus, edit mode    N  Alternate input mode
 N  Generate UPPERCASE menu           N  PRINT menu to DISK instead of printer
 N  Create DSECT file from menu       N  Create OBJECT TEXT deck from menu

Marker Characters: (characters or hex)
 ¦   Unnamed fields       ~     Named fields      `    Null characters

Field Naming Method: (Y|N)
 N  Use REXX names (AA.1, AA.2...)    N  Prompt for names after XMEDIT
 N  AUTONAME all fields               N  AUTONAME all Unprotected fields
    If AUTONAME  Truncate leading zeros N   Use Prefix        Start Number 1
                 And only name fields between          and
                      Press ENTER to continue; PF01/13 for HELP; PF03/15 to EXIT
```

Figure 4. XMEDIT Options Screen

Aside from the title lines, the boldface portions of the screen show the default options. Some of these defaults are based on the kind of terminal you are using, so the values on your screen may not match those in the figure. These defaults may be suitable for the menu you are creating; if so, you need only to supply a filename for your menu. Otherwise, modify the options as necessary, basing your choices on the following:

- The intended environment for your menu
- How you want your menu to behave
- Your personal work style
- Needed output from the menu

To make permanent modifications to any of the options, see Appendix F, "Changing the defaults by editing PROFILE XMEDIT" on page 127.

Completing the XMEDIT Options Screen

Complete the options screen to give XMEDIT some basic information about your prospective menu.

Use the TAB key to jump forward from field tq field on the XMEDIT Options Screen, the BACK TAB key to jump in reverse order, and the RETURN key to jump down the fields on the left side of the screen.

1. Type a filename for the menu. If you do nothing else on this screen, you **must** enter a menu name.
2. If you want to change any of the defaults, move the cursor to other options.
3. Change any of the defaults by typing over them.
4. Get HELP, if you need it, by pressing PF1 or PF13.
5. Press ENTER when you are finished with this screen.

Following are brief descriptions of the options on this screen. The descriptions are grouped below as they are grouped on the options screen, and are presented left to right, top to bottom. If you find that you need more detail about an option, look for it in "Notes on the XMEDIT options" on page 13. Because you need only enter a menu name in the first field on this menu, you may want to skip this section, too, until you need more detail about some of the options. To continue with the menu creation process go to "Typing the text for your menu in Input mode" on page 25.

Naming and profile options

```
---------------- XMENU 2  XMEDIT 3.00 Menu Generator/Editor -----------------
              * * * Copyright 1981,1989 Relay Technology * * *
 Menu Filename                Library name          - The menu to be edited.
USING Filename                Library name          - Menu to use as prototype.
Use Profile  PROFILE  - Profile macro executed at start; clear if NOPROF wanted
```

Option	Description
`Menu Filename`	CMS filename of the menu you want to create or edit. Each menu is stored in its own file on your A-disk and has filetype MENU. You **must** enter a valid CMS filename in this field before XMEDIT will transfer you to Input mode.
`Library name`	The filename of the menu library in which an existing menu is located that you wish to edit. You can use library files from any disk you have accessed. Menu libraries are created with the XMENULIB utility, which is introduced briefly in this chapter in "Use an existing menu as a template, and use menu libraries" on page 14 and described in detail in Part 2, The XMENU editor utilities reference, Chapter 4, "XMENULIB" on page 107.
`USING Filename`	CMS filename for an existing menu to be used as a template for a new menu you are creating. You can use a menu from any disk you have accessed. XMENU gives you a copy to modify, naming it whatever you specified in the `Menu Filename` field. For more details,

see "Use an existing menu as a template, and use menu libraries" on page 14.

`Library name` The filename of a menu library that contains the menu you wish to use as a template while creating a new menu. You can use library files from any disk you have accessed. Menu libraries are created with the XMENULIB utility, which is briefly introduced in this chapter in "Use an existing menu as a template, and use menu libraries" on page 14, and described in detail in Part 2, The XMENU editor utilities reference, Chapter 4, "XMENULIB" on page 107.

`Use Profile` The default profile for XMEDIT is provided in a file called PROFILE XMEDIT. Its filename, PROFILE, is displayed in this field. The profile defines the PF keys you use in XMEDIT and specifies some of the defaults you see highlighted on the XMEDIT Options Screen as well. You can create an XMEDIT file with another filename, if you wish, and can execute it by typing its filename over the word PROFILE in this field. See Appendix F, "Changing the defaults by editing PROFILE XMEDIT" on page 127, for information on altering the defaults set in this file. If a profile does not exist or if you choose to use no profile by blanking out this field, the PF keys default to the settings in the PROFILE XMEDIT shipped with the product. A copy of this file is included in Appendix F, "Changing the defaults by editing PROFILE XMEDIT" on page 127.

Menu characteristics

```
Menu Characteristics: (Y|N - except for SIZE)      43  /  80  Size in rows/cols
 N  327x extended attributes          N  Set MDT for unprotected fields
 N  Set Skip for protected fields     N  Sound alarm on display
```

`Size in rows/cols`
Specifies the screen size for your menu in rows and columns. The default values match the screen size of your terminal. Modify these two values if you are creating a menu for a terminal with a larger or smaller screen size. See "Screen sizes" on page 15 for details on menu sizes.

`327x extended attributes`
Specifies whether terminals displaying your menu may have extended attributes, such as color. Valid values: `Y` (*Yes*) or `N` (*No*). The default value matches the capability of the terminal you are using. If your menu might be displayed on other terminal types, or if you choose to use extended attributes for any of your menu's fields, make sure this is set to `Y` (*Yes*). See "Extended attributes" on page 15 for details.

`Set MDT for unprotected fields`
Sets the Modified Data Tag (MDT) on unprotected fields created during this session with XMEDIT. The default is `N` (*No*). See "The Modified Data Tag" on page 17 for more information.

Set Skip for protected fields
: When set to Y (*Yes*), specifies that when your menu is used, the user's cursor will skip over fields you define as protected. This saves your users time, since protected fields cannot accept input. See "Extended attributes" on page 15 for more information on protected fields.

Sound alarm on display
: When set to Y (*Yes*), specifies that an alarm will sound whenever the menu is displayed. For more information, see "Sound or override a menu alarm" on page 18.

Work setting

```
Work Setting: (Y|N)
 N  Use old XMENU menus, edit mode   N  Alternate input mode
 N  Generate UPPERCASE menu          N  PRINT menu to DISK instead of printer
 N  Create DSECT file from menu      N  Create OBJECT TEXT deck from menu
```

Use old XMENU menus, edit mode
: When set to Y (*Yes*), you use the "old style" full-screen menus rather than use XMEDIT's SAA-style pop-up windows for assigning attribute values and field names. This provides compatibility with versions of XMENU prior to Version 2.1.0. See "Use OLDWAY full-screen menus" on page 18.

Alternate Input mode
: When set to Y (*Yes*), specifies that Alternate Input mode is to be used instead of Normal Input mode. This option is useful if your terminal cannot display the field-marking characters or if you are using 327*x* emulation and want to minimize terminal I/O. Differences in the two Input modes are slight; see "Alternate Input mode" on page 19 for details.

Generate UPPERCASE menus, edit mode
: When set to Y (*Yes*), converts the menu to all uppercase letters when you exit and save the menu. On subsequent editing, all the text will be in uppercase.

PRINT menu to DISK instead of printer
: Sends output of the PRINT subcommand to a disk file, ***menuname*** **LISTING**, instead of to your virtual printer. Specify Y (*Yes*) if your virtual printer is not a line printer. See "Send print output to a disk file" on page 20 for details on modifying this file for printing on another type of printer.

Create DSECT file from menu
: When set to Y (*Yes*), creates an assembler DSECT file, ***menuname*** **COPY**, containing a DSECT image of the menu output data. See "Create an assembler DSECT file" on page 19 for more details.

Create OBJECT TEXT deck from menu
: When set to Y (*Yes*), creates an object deck from the menu, ***menuname*** **TEXT**, similar to the output of an assembler or compiler.

This can be loaded as part of your application module. See "Create an object deck" on page 21 for details on its use.

Marker characters

```
Marker Characters: (characters or hex)
 ¦   Unnamed fields      ~    Named fields     `   Null characters
```

`Unnamed fields`

The character used to mark the beginning of unnamed fields in Normal Input mode is a split vertical bar (¦). If you want to change this character, enter a new character's hexadecimal value (between X'41' and X'FE'), or simply type a character from the keyboard.

More information on the `Marker Characters` section of the XMEDIT Options Screen is provided in "Field-marking characters" on page 21.

`Named Fields` The character used to mark the beginning of named fields in Normal Input mode is a tilde (~). If you want to change this character, enter a new character's hexadecimal value (between X'41' and X'FE'), or simply type a character from the keyboard.

`Null characters`

The character used to display null (binary zero) characters in the menu, if there are any. If you want to change this character from its default (which is `, or X'79') enter a new character's hexadecimal number (between X'41' and X'FE'), or simply type a character from the keyboard.

Field naming method

```
Field Naming Method: (Y|N)
 N  Use REXX names (AA.1, AA.2...)   N  Prompt for names after XMEDIT
 N  AUTONAME all fields              N  AUTONAME all Unprotected fields
    If AUTONAME  Truncate leading zeros N   Use Prefix        Start Number 1
                 And only name fields between          and
                      Press ENTER to continue; PF01/13 for HELP; PF03/15 to EXIT
```

`Use REXX names (AA.1, AA.2...)`

When set to `Y` (*Yes*), requests that XMEDIT name fields using REXX-style compound variables for field names, for example, *AA.1*, *AA.2*, and so forth.

For more information on the `Field Naming Method` section of the XMEDIT Options Screen, see "Automatic field naming" on page 22.

`Prompt for names after XMEDIT`

When set to `Y` (*Yes*), requests that XMENU prompt you for the name of each field after you have finished creating the menu.

AUTONAME all fields
: When set to Y (*Yes*), requests that XMEDIT name each field, using the first four characters of the menu filename followed by a 3-digit field number, such as *XXXXnnn*.

AUTONAME all Unprotected fields
: When set to Y (*Yes*), requests AUTONAME for unprotected fields only.

Truncate leading zeros
: If you have set either of the AUTONAME options to Y (*Yes*), this option will allow XMEDIT to truncate leading 0s (zeros) from the 3-digit field numbers.

Use Prefix
: If you have set either of the AUTONAME options or REXX to Y (*Yes*), you can supply a prefix for field names here to be used instead of the first four characters of the menu name.

Start Number
: If you have set either of the AUTONAME options or REXX to Y (*Yes*), you can specify the number to be used to name the first field. If your application has other menus that have used AUTONAME, specify a number beyond the range used for those menus.

And only name fields between and
: You can specify a range of fields to name using the AUTONAME naming format (*XXXXyyy*). Enter the lower-numbered field name to the right of between and the higher-numbered field name to the right of and. Both fields entered are included in the range to be automatically named.

PF keys used for selecting options

The following PF keys are available from the XMEDIT Options Screen:

PF keys for selecting options

Key	Purpose
PF1/PF13	Displays the XMEDIT HELP screen of all XMEDIT subcommands.
PF2/PF14	Exits from XMEDIT.
PF3/PF15	Exits from XMEDIT.
CLEAR	Has no XMEDIT function; however, any input entered before pressing CLEAR is lost.
ENTER	Accepts the options as displayed, and transfers to Input mode.

Notes on the XMEDIT options

This section contains more detail on the options you can set for creating or editing a menu. If you are using the defaults or have made a few minor changes, you may want to skip this section until a time when you need to have more detail about some of the options. To continue with the menu creation process go to "Typing the text for your menu in Input mode" on page 25.

Each field on the XMEDIT Options Screen has a corresponding XMEDIT command line option name. These option names are included in this section, along with their Option Screen field prompts, to help familiarize you with using XMEDIT options with the the XMEDIT command as an alternative to using the XMEDIT Options Screen, and to familiarize you with the various option names. As you grow more familiar with XMEDIT and the default options it sets, you may choose to bypass the Options Screen by specifying one or more options directly on the XMEDIT command line. For example, instead of just typing the command XMEDIT, you might type the following command:

```
XMEDIT NEWMENU (SKIP DISK SIZE 24 80 EXT
```

Although you will probably use the XMEDIT Options Screen to specify such options for your menu, we'll explain this example in a bit more detail to familiarize you with adding options to the XMEDIT command. In this example, the XMEDIT command will either create a new menu or edit a previously saved menu named NEWMENU using the options specified after the left parenthesis. These options will be accepted as additions to or replacements for the default options provided by PROFILE XMEDIT (for more information on this file and the default XMEDIT environment, see Appendix F, "Changing the defaults by editing PROFILE XMEDIT" on page 127). The options used in this example are briefly described below:

SKIP	The user's cursor will skip over all protected fields on the menu.
DISK	When PA2 is pressed from XMEDIT Display mode, a file called NEWMENU LISTING A will be created that includes a "screen print" of the menu plus a descriptive listing of each field's location, attributes, and name, if specified.
SIZE 24 80	NEWMENU's maximum size is 24 lines long by 80 columns wide.
EXT	If assigned, extended attributes will be shown on terminals with the capability to display them.

Each of these options and others are described below in more detail.

Use an existing menu as a template, and use menu libraries

When you design menus for an application, you may find that many of the menus will be similar in many respects. With XMEDIT you can save time and effort when you are creating several similar menus by copying an existing menu and modifying it to suit your needs.

To use an existing menu as a prototype for a new menu you are creating:

1. Enter a filename for the new menu you want to create in the Menu Filename field.
2. Enter the filename of the menu you want to copy in the space following the words USING Filename.

   If the menu you are copying resides in a library (a file with the filetype XMENULIB that is created with the XMENULIB utility), enter the library name in the space following the words Library name across from Using filename.

Note: The library can be on any accessed disk. You can alternately specify LIB *libname* as a command line option when invoking XMEDIT. The XMENULIB utility and the LIB option are discussed in detail in Part 2, The XMENU editor utilities reference, "XMEDIT options" on page 56 and Chapter 4, "XMENULIB" on page 107.

3. Press ENTER.

   XMEDIT retrieves the existing menu and displays it in Input mode. Transfer to Edit mode and you can easily modify this menu. For more information on editing menus, see "Rearranging the text in Edit mode" on page 28.

Screen sizes

The size of the menus you create is by default the same size as the terminal on which you are creating them. If the terminal on which you are creating menus is larger than the terminals on which the menu will be displayed, you should set the size for the smallest terminal.

You can change the size of a menu by specifying the number of rows (or lines) and the number of columns on the XMEDIT Options Screen. Simply change the number of rows and the number of columns in the fields preceding `Size in rows/cols`. The first number is for rows, and the second for columns.

The minimum size of a menu is one row by one column. The maximum size of a menu is the product of rows and columns, not to exceed 16,384 bytes. For example, a 64 column menu can span 256 rows.

If you change the values in this field when you are copying an existing menu (with `USING filename`), the copied menu is set to the size you specify before being displayed. If the copied menu is larger than the new one, lines are lost.

You can use the XMEDIT command line option SIZE to specify a different menu size than your terminal screen size. Menus can also be sized dynamically with the MENUEXAM utility. Called from within a program that displays menus, MENUEXAM dynamically places lines in a menu to fill any terminal screen size. For more information on the SIZE option see Part 2, The XMENU editor utilities reference, "XMEDIT options" on page 56. For information on the MENUEXAM utility see the *XMENU Utilities Reference*.

Extended attributes

You can create menus that use 327*x* extended attributes for terminals that support these capabilities. You can use extended attributes even if the menus might be used on terminals that do not support them. For example, you might highlight some information on a menu with one of the extended colors, such as yellow. If the menu is displayed on a non-color terminal, the field will not appear yellow, but neither will an error occur: XMENU just ignores the extended attributes.

Before XMEDIT displays the XMEDIT Options Screen, it checks the terminal you are using to determine whether it supports extended attributes, and then puts a `Y` or a `N` in the `327x Extended Attributes` field, depending on what it found. You can change the setting, if you want, by entering a `Y(Yes)` or `N (No)` in this field.

Note: If you enter XMEDIT options from the command line and want extended attributes enabled, you must include EXT on the command line even if your terminal has extended capabilities. This is true unless you have previously edited the menu with EXT set. If you are invoking XMEDIT to edit a previously created menu, the attribute type previously set for that menu overrides your terminal type. Menus with extended attribute data cannot be converted back to non-extended format, but they can be displayed on terminals without extended attributes. Menus with non-extended format may be changed to extended attribute format at any time.

If you always want to use extended attributes, you can include a line, `SET EXT ON`, in a PROFILE XMEDIT file to specify that you are creating menus with extended attributes. For more information on PROFILE XMEDIT files, see Appendix F, "Changing the defaults by editing PROFILE XMEDIT" on page 127.

There are two different kinds of field attributes, basic and extended:

Basic attributes fall into four groups:

1. Intensity

   You can make a field **BRIGHT**, normal intensity (**DIM**), selector pen detectable (**LGHTPEN**), or blacked out completely (**DARK**) (for password fields, for example).

2. Protection

   You can make a field protected (**PROT**) so users cannot enter data into it, unprotected (**UNPROT**) so users can enter data into it, or numeric (**NUMERIC**) so users can only enter numeric data.

   Note: Some terminals may not support numeric fields. Pressing the SHIFT key will allow the user to override numeric attributes.

3. Skip control (**SKIP**)

   You can have XMEDIT skip the cursor over protected fields. For more information, see "Set SKIP for protected fields" on page 17.

4. Modified Data Tag (**MDT**)

   This attribute is only occasionally changed from the default; that is, MDT is usually set OFF for all fields (NOMDT). With the default setting in effect, whenever users enter data into a field, MDT is automatically changed to ON. You can have XMENU set MDT ON for a field at each use of the menu, regardless of whether the user enters data into any fields. For more information on MDT, see "The Modified Data Tag" on page 17.

Extended attributes include these groups:

1. Color

   The extended attribute of color can itself include either basic or extended attributes. Basic colors are red, green, blue, and white. Fields not assigned color attributes default to these four colors. Extended colors include yellow, turquoise, and pink, as well as red, blue, green, and white.

2. Extended highlighting

The extended highlighting types are underscore (**UNDERSCORE**), reverse video (**REVERSE**), blinking (**BLINK**) and default (**DEFHI**). The default extended highlighting type depends on the output terminal type.

3. Programmed symbol sets (**PS**)

   With some hardware, you can load alternate character sets from the mainframe into the terminal, for example Katakana or Hebrew characters. Programmed symbol sets are also referred to as logical symbol sets.

While familiarizing youself with the possible field attributes, you may find the listing of basic and extended 327*x* attributes presented in Appendix C, "Attribute types and values" on page 121 helpful.

The attributes you assign to a field should be appropriate to the field's function. Protected fields are typically used for data which you show to the user but which cannot be changed by the user. Unprotected fields are used to get input to the application from the user. Field brightness, color, highlighting, and symbol sets are used for special emphasis or special purpose; for example, normal brightness is used for the majority of fields, but error message fields might stand out more with bright or blinking attributes. Password fields and fields that contain other sensitive data should be invisible.

Set SKIP for protected fields

You can have XMEDIT skip the cursor past each protected field to the next unprotected field on the menu. This saves the user time since the cursor does not have to be manually moved past fields that do not accept user input.

To skip the cursor past protected fields enter a `Y (Yes)` in the field preceding `Set Skip for protected fields` on the XMEDIT Options Screen, or include SKIP as an option of the XMEDIT command.

When you use SKIP, only the fields created or modified **during this XMEDIT session** will have SKIP set for them. Fields created on the menu during an earlier session are not affected.

The Modified Data Tag

You can have XMENU set the Modified Data Tag (MDT) ON for unprotected fields, although the default, most frequently used setting is OFF. With the default setting in effect, whenever users enter data into a field, MDT is automatically changed to ON. You can have XMENU set MDT ON for unprotected fields at each display of the menu, regardless of whether the user enters data into any fields. This results in the fields being "primed" with input upon the menu's display. For more information on MDT, see Part 2, The XMENU editor utilities reference, "XMEDIT options" on page 56.

When you change the XMEDIT MDT option, or enter a `Y (Yes)` in the field preceding `Set MDT for unprotected fields` on the XMEDIT Options Screen, only the fields created or modified **during this XMEDIT session** will have MDT set ON. Menu fields created during an earlier session are not affected.

Sound or override a menu alarm

To cause the alarm to sound when a menu is displayed, enter a `Y (Yes)` in the field following `Sound alarm on display`, or include ALARM as a command line option when invoking XMEDIT. You can use the XMEDIT command line option NOALARM to clear an ALARM option that might have been set in a previous XMEDIT editing session. See Part 2, The XMENU editor utilities reference, "XMEDIT options" on page 56 for more information on using (ALARM) and overriding (NOALARM) the terminal alarm.

Use OLDWAY full-screen menus

In earlier versions of XMENU, prior to Version 2.1.0, full-screen menus were displayed whenever you assigned attributes or named a field from Input mode.

If you prefer to use the full-screen displays for these functions, enter `Y (Yes)` in the field preceding `Use old XMENU menus, edit mode` on the XMEDIT Options Screen, or include the OLDWAY option on the XMEDIT command line.

With OLDWAY, when PF6/PF18 is used to assign, review, or change a field's attributes, a full-screen menu appears that displays the attributes currently set for that field. See Figure 30 on page 124 for an example of this screen. Without OLDWAY, when you press PF6/PF18, an SAA-like action bar replaces the full-screen presentation (see Figure 19 on page 43). Refer to Appendix D, "Assigning attributes using the OLDWAY full-screen menus" on page 123, for information on using the full-screen method.

Also with OLDWAY in effect, a full-screen display is presented when you press PF12/PF24 from Input mode to assign a field a name (see Figure 31 on page 125). Without OLDWAY, a pop-up window (see Figure 22 on page 46) appears in place of the "old" full-screen menu for naming fields. Refer to Appendix E, "Naming fields by using the OLDWAY full-screen menus" on page 125 for information on using the full-screen method.

If you use the XMENU command in place of the XMEDIT command, the OLDWAY option is implicitly set, and there are other cosmetic differences in the working environment.

Create an all-uppercase menu

To create a menu with uppercase letters enter a `Y (Yes)` in the field preceding `Generate UPPERCASE menu` on the XMEDIT Options Screen. This option can be directly invoked if you use the UPCASE command line option when you first enter the XMEDIT command.

When you use this option, letters still appear in upper- and lowercase when you are entering text; however, when you exit and save the menu, XMEDIT converts all the letters to uppercase. The next time you edit the menu, all the text will be in uppercase.

Create an assembler DSECT file

You can create an assembler DSECT after creating a menu. You can use a DSECT for high-performance or special-purpose applications, such as those that use I/O systems other than those provided by some of XMENU's subroutines. An assembler DSECT file named ***menuname*** **COPY** is created containing an assembler DSECT image of the menu output data.

Assembler DSECT files have the following characteristics:

- There is an EQU symbol named ***menuname*****S**, which contains the length of the DSECT in bytes. If the menu name is eight characters long, the last character is overlaid by **S**.
- Fields within the DSECT are labeled with the field names created during menu creation or modification.
- Lines in the DSECT which contain attributes have comments describing the type of attribute.
- The header line contains the line and column of the initial cursor position.
- Each line of the data contains its line and column position.
- All data is defined with DC statements, so the file can be used to define constants by changing the DSECT statement to a CSECT statement.

The ***menuname*** **COPY** file is created when either the MENTEXT or COPY option is specified on the XMEDIT command line, or when you enter `Y (Yes)` in the field preceding `Create DSECT file from menu` on the XMEDIT Options Screen.

Alternate Input mode

XMEDIT lets you enter text in two modes: Normal Input mode and Alternate Input mode. Alternate Input mode is used for environments that cannot display the field marker characters for named and unnamed fields (~ and ¦ respectively), or that benefit from reduced terminal I/O, such as 327*x* emulators like SIM3278. You can change the special characters used by XMEDIT to mark fields. For more information see "Field-marking characters" on page 21.

Alternate Input mode differs slightly from Normal Input mode:

- In **Normal** Input mode, the entire input area is one unformatted field.
  - As you enter fields, XMEDIT places special characters on the screen to show you where the fields are.
  - You locate a field by looking for the special character that marks the start of a field.
  - You remove a field by typing over the special character that marks the start of a field.

    Caution: If you use the 327*x* INSERT or DELETE keys anywhere on your menu, you may relocate the special characters and destroy all the fields. Relocating a special character destroys the field it marks.
- In **Alternate** Input mode, the fields you create are placed on the screen as true 327*x* fields. XMEDIT places a 327*x* attribute character, rather than a

field-marking character, on the screen and sets the field to unprotected so that you can continue to enter data or more fields.

- You locate the start of a field (which appears unmarked) by using the forward or backward tab keys. These keys actually move the cursor to the character after the attribute; you must move the cursor one position backward to be directly over the start of the field.
- You remove a field by placing the cursor on the attribute space at the start of the field and pressing the PA2 key, which is the default DESTROY key in Input mode.

  If the cursor is not on the start of a field when the field delete key is pressed, XMEDIT moves the cursor to the start of the preceding field.

All other XMEDIT functions work the same in Normal and Alternate Input modes.

To use Alternate Input mode, include ALT as a command line option when invoking XMEDIT, or, from the XMEDIT Options Screen, enter `Y (Yes)` in the field preceding `Alternate Input mode`.

You can also use SET ALT ON in a PROFILE XMEDIT file to always have XMEDIT enter Alternate Input mode. For more information on this file, see Appendix F, "Changing the defaults by editing PROFILE XMEDIT" on page 127.

Send print output to a disk file

After you create a menu you might find it helpful to have a printed copy of it. For information on printing a menu, see "Print the menu" on page 49.

By default, XMEDIT sends print output to your virtual printer. This output is formatted for printers, such as IBM 1403 printers, that can recognize ANSI (ASA) control characters in record column one.

If you do not have a 1403-type printer, you can route the print output to a disk file and edit it so that it will print on your printer. When you edit the file you will have to remove the carriage control characters from column one and change any highlighted text from triple overstrikes to whatever is appropriate for your printer.

To send a copy of a menu to a disk file, enter a `Y (Yes)` in the field following `PRINT menu to DISK instead of printer` on the XMEDIT Options Screen, or include the DISK option on the XMEDIT command line. When you execute a print command, the print output goes to a file named ***menuname*** **LISTING**, rather than to your virtual printer.

If you generate documentation in DCF (SCRIPT/VS) format and want to include "screen prints" of your menus, use the XMEDIT option SCRIPT to have the PRINT command create a SCRIPT output file.

Create an object deck

To create a TEXT deck for a menu from the XMEDIT Options Screen, enter `Y` (*Yes*) in the field preceding `Create OBJECT TEXT deck from menu`. A ***menuname*** **TEXT** file is also created if either the MENTEXT or TEXT option is specified on the XMEDIT command line.

An object file is similar to the output of an assembler or compiler. You can load an object deck as part of your application module and either pass its in-storage addresses to the XMENU subroutine MLOAD, or use it as part of a non-XMENU application. You can use an object deck for high-performance or special-purpose applications, such as those that use I/O systems other than those provided by some of XMENU's subroutines.

There are four external reference names in the object deck:

menuname Points to the first byte of the loaded file.

This is the name by which the file is loaded.

menunameL Points to a fullword containing the length in bytes of the output data. If the menu name is eight characters long, the last character is overlaid by **L**.

menunameI Points to the data contained in the first record of a menu file. If the menu name is eight characters long, the last character is overlaid by **I**.

This address would be passed to MLOAD as the third parameter to load an in-storage menu.

menunameO Points to the actual output image which is contained in the second record of a menu file. If the menu name is eight characters long, the last character is overlaid by **O**.

This address would be passed to MLOAD as the fourth parameter to load an in-storage menu.

Field-marking characters

When you create a field in Normal Input mode, XMEDIT places special characters on the menu to show you where the field is. XMEDIT marks fields with two different characters:

- Unnamed fields are marked with a split vertical bar (¦, hexadecimal 6A)

  For information on defining a field in Input mode by pressing PF6/PF18 to assign attributes, see "Define fields by assigning attributes" on page 41.

- Named fields are marked with a tilde (~, hexadecimal A1)

  For more information about naming fields by pressing PF12/PF24 in Input mode, refer to "Name a field" on page 45, and alternately, for information on having XMEDIT name fields automatically, see "Automatic field naming" on page 22.

The following menu, displayed in Input mode, shows one named field (the `Manual Number:` field); all other fields are unnamed:

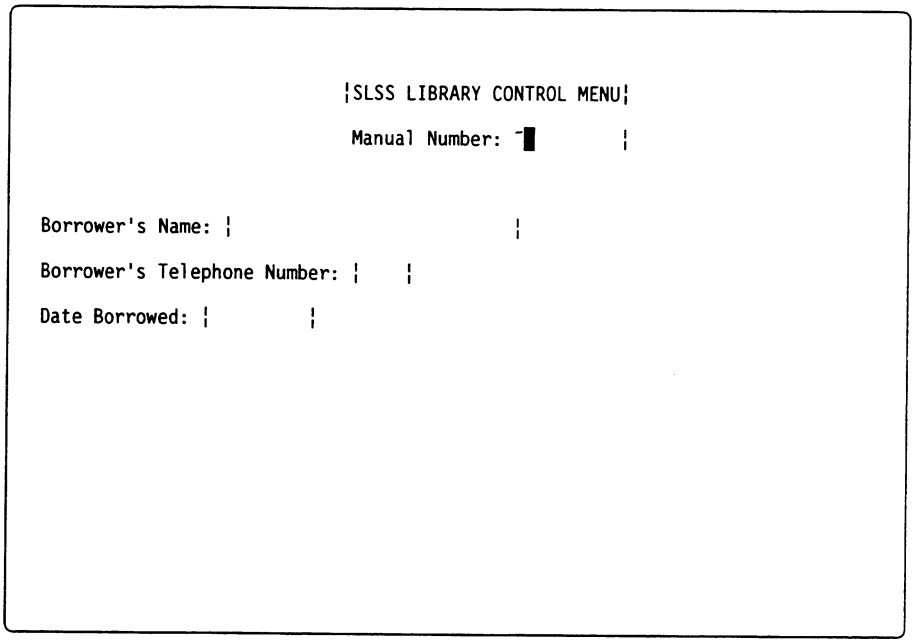


Figure 5. Sample menu with one named field

If your terminal does not display one or both of these characters, or if it translates them into different characters, you can change the field markers. The characters to be used can be changed on the XMEDIT Options Screen by simply typing the desired character into the appropriate field, or by entering the desired character's hexadecimal value into the field. If you choose to enter a hexadecimal value, you must choose characters greater than hexadecimal 40 and less than hexadecimal FF.

You can use the XMEDIT option CHAR to change the default field markers, or you can change the values with the XMEDIT subcommand SET CHAR *xx xx* in the PROFILE XMEDIT file. See Part 2, The XMENU editor utilities reference, "XMEDIT options" on page 56 for more information on the CHAR option, and see Appendix F, "Changing the defaults by editing PROFILE XMEDIT" on page 127, on changing these characters in the PROFILE XMEDIT file.

Automatic field naming

The job of naming fields can be done in several ways with XMEDIT. Refer to "Name a field" on page 45 for basic information on manually naming fields while in Input mode. From the XMEDIT Options Screen you can specify that XMEDIT use an automatic method for assigning names to fields:

- **Prompt** you for field names as you leave an XMEDIT session (the NAME option)
- Automatically name **all** fields (the AUTONAME option)
- Automatically name only **unprotected** fields (the NAMEUNP option)
- Automatically name fields using **REXX-type** names (the REXX option)

Telling XMEDIT to prompt you for field names: You can tell XMEDIT to prompt you for the name of each field on the menu after you have created the menu. When you have finished creating the menu, XMEDIT prompts you in turn for each field name. Type the name in the space provided, or leave it blank to keep a field un-named, and press ENTER.

To have XMEDIT prompt you for field names, enter a `Y (Yes)` in the XMEDIT Options Screen field preceding `Prompt for names after XMEDIT`.

Note: If you tell XMEDIT to prompt you for field names, you can still name fields as you create them using the Name-a-field pop-up menu. For more information, see "Name a field" on page 45.

This prompting function can also be invoked from the XMEDIT command line, by including the NAME option.

Automatically naming all menu fields: You can tell XMEDIT to automatically name **all** the fields on a menu after you have created it. From the XMEDIT Options Screen, place a `Y (Yes)` in the field preceding `AUTONAME all fields`. This function can also be invoked from the XMEDIT command line by including the AUTONAME command line option.

If you tell XMEDIT to name all the fields, XMEDIT writes over any existing field names.

XMEDIT assigns names to fields according to the order in which they appear on the menu. Field names are of the form *NNNNxxx*:

- *NNNN* are the first four characters of the menu name
- *xxx* is a three-digit number, beginning with 001

For example, if you had a menu named MYMENU, the first autonamed field would be MYME001, the second MYME002, etc.

When you have XMEDIT automatically name menu fields by using AUTONAME, NAMEUNP, or REXX (see the next sections for information on NAMEUNP and REXX), you can also tell XMEDIT to suppress leading zeros, use specific names and numbers for fields, and name only a range of fields:

1. Suppress leading zeros in the field names (the TRUNC option)

   For example, if you had a menu named MYMENU, the first field would be named MYME1, the second MYME2, etc. Suppressing leading zeros makes it easier to use XMEDIT-assigned field names in EXECs using indexed EXEC variables. Use the TRUNC option on the XMEDIT command line or enter a `Y (Yes)` in the field following `Truncate leading zeros` on the XMEDIT Options Screen.

2. Use a particular one-to-four-character prefix when naming fields (the FNAME option)

   Rather than use the default first four characters of the menu name, you can specify a prefix of one to four characters using FNAME. FNAME is useful for assigning the same names to groups of fields in different menus. Use the FNAME option on the XMEDIT command line or enter a name in the field following `Use prefix` on the XMEDIT Options Screen.

3. Use a particular starting number when naming fields (the FNUMB option)

   Rather than starting with 001, which is the default, you can use FNUMB to set a specific starting number. This can be helpful when you have an application displaying several menus whose menu names, or filenames, begin with the same four characters and you do not wish to use the FNAME option. Use the FNUMB

option on the XMEDIT command line or enter a number in the field following `Start Number` on the XMEDIT Options Screen.

4. Name only fields between and including a start field and an end field (the FGROUP option)

   Remember, XMEDIT assigns names to fields according to the order in which they appear on the menu. For example, you can tell XMEDIT to only name the second through the fifth fields on a menu that has seven fields by specifying FGROUP *second-field-name fifth-field-name*. If you only specify a starting field, XMEDIT names every field from the starting field to the end of the menu. Use the FGROUP option on the XMEDIT command line or enter a *start-field-name* and an *end-field-name* in the fields following `And only name fields between` on the XMEDIT Options Screen. You must name both the starting-field and ending-field to use this option.

You **must** use either the AUTONAME (`AUTONAME all fields`), NAMEUNP (`AUTONAME all Unprotected fields`), or REXX (`Use REXX names (AA.1, AA.2. . .)`) options to use the TRUNC, FNAME, FNUMB, and FGROUP command line options.

Automatically naming only unprotected fields: You can have XMEDIT automatically name only unprotected fields by entering a `Y (Yes)` in the field preceding `AUTONAME all Unprotected fields` or by including the NAMEUNP option on the XMEDIT command line.

With NAMEUNP you can also specify any of the XMEDIT options that were introduced in the AUTONAME section above:

TRUNC To suppress leading zeros in the field names

FNAME To specify a one-to-four-character prefix when naming fields

FNUMB To use a particular starting number when naming fields

FGROUP To name only fields between and including a *start-field-name* and an *end-field-name*

You can also use NAMEUNP in combination with REXX to name unprotected fields with REXX-type names.

Automatically naming all menu fields using REXX-type names: You can tell XMEDIT to automatically name fields with REXX-type names. REXX-type names are of the form ***XXX.yyy***, where:

- ***XXX*** is a REXX variable stem
- ***yyy*** is a leading-zero-suppressed ascending number

For example, if you had a menu named MYMENU and specified REXX, the first field would be named MYM.1, the second MYM.2, and so on. The period between the ***XXX*** and the ***yyy*** allows the use of REXX variable arrays.

The fields are not numbered 001, 002, etc. because REXX-type names are automatically and implicitly leading-zero suppressed. In other words, when you set the REXX option, you also implicitly set TRUNC and AUTONAME.

You can name all fields (the default) with REXX-type names, or only unprotected fields:

- If you are using the XMEDIT Options Screen and want to name all fields with REXX-type names enter a `Y (Yes)` in the field preceding `Use REXX names (AA.1, AA.2. . .)` Alternately, you can use the REXX command line option when invoking XMEDIT.
- If you are using the XMEDIT Options Screen and want to name only unprotected fields with REXX-type names, enter a `Y (Yes)` in the field preceding `Use REXX names (AA.1, AA.2. . .)` and enter another `Y (Yes)` in the field preceding `AUTONAME all Unprotected fields`. Alternately, you can include the REXX and NAMEUNP options on the XMEDIT command line. For more information on NAMEUNP, see the previous section.

Additionally, the options FNAME, FNUMB, and FGROUP can be used with REXX as they can with all the automatic naming options.

Modify XMEDIT's default PF key settings

The PF keys in XMEDIT's Input and Edit modes have default settings that you can keep or modify to suit your needs. To modify the defaults permanently, see Appendix F, "Changing the defaults by editing PROFILE XMEDIT" on page 127. To modify the Input mode defaults for the duration of an XMEDIT session, simply type over the PF key definitions on the screen that is presented when you press PF13 from Input mode.

You can swap PF key assignments if you prefer using one PF key in XMEDIT over another for a frequently-used function, or you can assign very specific functions, tailored for a particular menu, to a PF key. For example, if there are many fields that will share the attributes UNPROTected, RED, and BLINK, you can type all these attribute commands over an existing PF key assignment. Then you can use that PF key to speed the process of assigning those three attributes to the desired fields.

Typing the text for your menu in Input mode

If you enter the XMEDIT command without specifying a menu name or any options, and if your terminal's screen is at least 24 lines long, the XMEDIT Options Screen is presented to prompt for a menu name and any special options you may need or desire.

After you complete the XMEDIT Options Screen and press ENTER, you see the screen shown in Figure 6 on page 26.

```
                  ** Welcome *************************
                  *                                  *
                  *    Welcome to the XMEDIT menu    *
                  *  creation/modification program.  *
                  *     You are about to create a    *
                  *     new menu. Press ENTER to     *
                  *  continue, PF03 (PF15) to exit.  *
                  *                                  *
                  ************************************

After pressing ENTER, you will be presented with a blank screen.  This
enables you to create a menu/panel  in "WYSIWYG" (What you see is what
you get) manner.

If  you are  using  the  default PROFILE  XMEDIT,  pressing PF01  will
present a  command line.  From the  command line,  you can  issue any
XMEDIT subcommand.  For example: "QUERY SCREEN INPUT" will display the
current PF Key settings.

Press ENTER now to continue your XMEDIT session.
```

Figure 6. XMEDIT Welcome Screen

Moving from mode to mode: As mentioned earlier, you can transfer among three modes while creating a menu: Input, Edit, and Display. Once you press ENTER to clear the Welcome Screen, you will be in Input mode, where you can type the text for your menu.

Note: If you press ENTER again, you shift to Edit mode, where the text area is protected from input and where you use XEDIT-like prefix area commands to perfect the text layout. Press PF3/PF15 to return to Input mode. You can shift back and forth between these two modes until your menu is exactly as you desire.

Using Input mode to enter text

You first use Input mode to enter the text you want to appear on your menu. Follow these basic instructions to enter, use, and exit Input mode:

Steps for entering text in Input mode

1. Press ENTER to clear the Welcome Screen and display a totally blank Input mode screen.
2. Move the cursor with the following keys to position it for typing:
   - The four cursor movement keys (↓↑←→)
   - The TAB key
   - The RETURN key (**not the ENTER key**)
3. Type text for your menu more or less as you would like to see it displayed. Don't worry about centering the text perfectly, or positioning it on the line exactly as you would like it to appear; you will be able to perfect these elements of your menu design in Edit mode.

   Note that the entire screen is one big field at this point. Therefore, **do not** use the following keys to modify your text while in Input mode:

   - INSERT
   - DELETE
   - ERASE EOF

   These keys will cause the Input mode screen field to "wrap" with undesirable results.
4. Press ENTER to transfer to Edit mode. If you do this by accident, press PF3/PF15 to return to Input mode.

You should not need the other PF keys at this point. They are described fully in "Completing the menu from Input mode" on page 38, where you will be using them.

Figure 7 shows a sample menu during Input mode. Although the menu is not laid out perfectly at this point, the once-blank screen now contains the text required.

```
                COMPUTER ACCOUNT REQUEST

Fill in the following fields and press PF12 or PF24.
Another screen will be displayed where you can specify
your computer account needs.


NEW EMPLOYEE INFORMATION


 Full Name          ==>
 (with middle initial)

 Employee Number    ==>

 Userid             ==>            Password        ==>

 Department Number  ==>            Mail Stop       ==>

 Office Number      ==>            Phone Extension ==>


Press PF3 or PF15 to QUIT
```

Figure 7. Sample menu during Input mode

Notice that the text for the menu's title, general information about the menu, field prompts, and PF key definitions are all typed on the screen in the approximate positions desired for this sample menu. You could type any variety of titles, field prompts, PF key definitions, and command lines, for example, on your menu. What you choose to type on your menu while in Input mode should be based on your expectations of what will be done with your menu by the user and the supporting application program.

You can, to use a simple example, design a menu with a title, a list of choices, a PF key number next to each list item, and instructions to the user on how to make a selection. The application program supporting this menu would display the menu, wait for the user to press a PF key, trap the PF key that was pressed, and associate it with the appropriate subroutine. If you wanted to let your users enter commands, simply type a command line as you would like it to appear on your menu, and support its use in your program. In other words, consider Input mode as a blank slate where your ideas and needs can take whatever form is desired or required.

Rearranging the text in Edit mode

Edit mode enables you to rearrange the parts of your menu. You enter Edit mode by pressing ENTER from Input mode. Figure 8 shows the sample menu as it would look in Edit mode.

```
1                 COMPUTER ACCOUNT REQUEST
2
3  Fill in the following fields and press PF12 or PF24.
4  Another screen will be displayed where you can specify
5  your computer account needs.
6
7
8  NEW EMPLOYEE INFORMATION
9
10
11   Full Name          ==>
12   (with middle initial)
13
14   Employee Number    ==>
15
16   Userid             ==>            Password        ==>
17
18   Department Number  ==>            Mail Stop       ==>
19
20   Office Number      ==>            Phone Extension ==>
21
22
23
24 Press PF3 or PF15 to QUIT
```

Figure 8. Sample menu during Edit mode

Note that the editing screen has a *prefix area* containing line numbers. This prefix area looks and functions much like the prefix area in XEDIT.

Using the editing commands

You edit a menu by typing editing commands in the prefix area to add, delete, move, copy, shift, justify, and/or center lines on your menu. The text area of your menu is protected during Edit mode: **you cannot type in the text area while in Edit mode.** If you need to change or place additional text on your menu, return to Input mode. While in Edit mode you can do the following things:

- Type any of the editing commands in the prefix area, pressing ENTER to invoke them. A description of all the commands follows.
- View a HELP display of the editing commands by pressing PF1/PF13.
- Scroll toward the right (to shift the text left) by pressing PF11/PF23 if you want to see the rightmost four columns of your menu; the menu has been shifted four columns to the right to make room for the prefix area. To scroll left again, press PF10/PF22.
- Press PF3/PF15 when you are ready to return to Input mode, where you can enter more text or start defining your fields.

Prefix area editing commands

Command	Purpose
CE	Centers the current line.
LE	Left-justifies the current line, making it flush with the left margin of your menu.
RI	Right-justifies the current line, making it flush with the right margin of your menu.
*n*L **or** *n*<	Shifts data on the current line to the left *n* spaces.
*n*R **or** *n*>	Shifts data on the current line to the right *n* spaces.
*n*E	Expands the line by inserting *n* blanks at the cursor position.
*n*S	Shrinks the line by removing *n* characters at the cursor position.
A, *n*A, I, **or** *n*I	Inserts a blank line or *n* blank lines below the current line.
D, *n*D, **or** DD	Deletes the current line or *n* lines including the current line, or deletes a block of lines, where DD marks the first and last lines of the block.
C, *n*C **or** CC	Copies the current line or *n* lines including the current line to a marked destination, or copies a block of lines, where CC marks the first and last lines of the block. Type F or T for *Following* or *To*, or type P or B for *Preceding* or *Before* on the appropriate line at the destination.
M, *n*M, **or** MM	Moves the current line or *n* lines including the current line, or moves a block of lines to a marked destination, marked with F or T, P or B, as explained directly above.
" **or** ""	Duplicates the current line or a block of lines immediately beneath the line or lines marked to be duplicated. "" marks the first and last lines of the block.
*n*G **or** GG	Imports or gets lines from a CMS file. See "Import lines from a CMS file (G, GG)" on page 34 for details.

"Notes on the editing commands" contains examples of editing commands with more details about them. If you do not need that information now, you can move past it to "Completing the menu from Input mode" on page 38.

PF key settings for Edit mode

The following PF keys will help you during Edit mode:

PF keys for Edit mode

Key	Purpose
PF1/PF13	Displays a HELP screen of prefix area editing commands.
PF3/PF15	Exits Edit mode and transfers back to Input mode.
PF10/PF22	Scrolls the menu to the left on the screen.
PF11/PF23	Scrolls the menu to the right on the screen.
ENTER	Executes prefix area editing commands.

Notes on the editing commands

This section contains more detail on the editing commands. Where appropriate, sample screens showing the editing commands typed in the prefix area of a menu are shown. Case is not significant when typing prefix area commands; uppercase and lowercase can be used interchangeably.

Add lines to your menu (A, I)

To insert additional blank lines into the menu, place the cursor on the number of the line after which the additional line is required. Enter "a" or "i". If more than one line is required, enter the number of lines immediately preceding the "a" or "i". For example, to add 3 lines, you would enter "3a" or "3i".

Be careful: XMENU does not allow you to exceed the number of lines you specified for your menu. When you add lines to a menu, lines on the bottom may "fall off" or be truncated from the bottom of the menu, and their fields and attributes, if previously assigned, will be lost.

To minimize the possibility of truncated lines, XMEDIT carries out all line delete operations before line add operations. Thus, if you want to add a line to the menu, but have a line you want to keep at the bottom of the menu, be sure to invoke the add and delete operations at the same time, or perform the line deletions before the line additions.

You can exceed the maximum number of lines as a result of inserting, adding, duplicating, copying, or getting lines from a CMS file.

In the following example, a menu set for a 24-line display has a prefix area command, "a", entered on line 1:

```
a 1 ------------------- SQL/DS Operator Command Selection Menu -------------------
  2                     Copyright 1989 Relay Technology, Inc.
  3
  4 Press ---------------------------- To display ---------------------------------
  5 PF01 SHOW ACTIVE
  6 PF02 SHOW BUFFERS
  7 PF03 Exit the SQLOPER menu.
  8 PF04 SHOW DBCONFIG
  9 PF05 SHOW DBEXTENT
 10 PF06 SHOW      DBSPACE      <-- enter DBSPACE number
 11 PF07 SHOW LOCK ACTIVE
 12 PF08 SHOW LOCK DBSPACE      <-- enter ALL or DBSPACE number (default "ALL")
 13 PF09 SHOW LOCK GRAPH    USERID           or AGENT
 14 PF10 SHOW LOCK MATRIX
 15 PF11 SHOW LOCK USER     USERID           or AGENT          (default "ALL")
 16 PF12 SHOW LOCK WANTLOCK USERID           or AGENT          (default "ALL")
 17 PF13 SHOW LOG
 18 PF14 SHOW SYSTEM
 19 PF15 Exit the SQLOPER menu.
 20 PF16 SHOW USERS
 21 PF24 COUNTER *
 22 ENTER Command here ->
 23
 24 --------------- Date: mm/dd/yy Time: hh:mm:ss USERID:    ----------------
```

Figure 9. Sample menu with edit command ready to add a line

Notice that the last line, which would have displayed a date, time, and userid, is truncated from this 24-line menu as a result of executing the "a" prefix command:

```
  1 ------------------- SQL/DS Operator Command Selection Menu -------------------
  2
  3                     Copyright 1989 Relay Technology, Inc.
  4
  5 Press ---------------------------- To display ---------------------------------
  6 PF01 SHOW ACTIVE
  7 PF02 SHOW BUFFERS
  8 PF03 Exit the SQLOPER menu.
  9 PF04 SHOW DBCONFIG
 10 PF05 SHOW DBEXTENT
 11 PF06 SHOW      DBSPACE      <-- enter DBSPACE number
 12 PF07 SHOW LOCK ACTIVE
 13 PF08 SHOW LOCK DBSPACE      <-- enter ALL or DBSPACE number (default "ALL")
 14 PF09 SHOW LOCK GRAPH    USERID           or AGENT
 15 PF10 SHOW LOCK MATRIX
 16 PF11 SHOW LOCK USER     USERID           or AGENT          (default "ALL")
 17 PF12 SHOW LOCK WANTLOCK USERID           or AGENT          (default "ALL")
 18 PF13 SHOW LOG
 19 PF14 SHOW SYSTEM
 20 PF15 Exit the SQLOPER menu.
 21 PF16 SHOW USERS
 22 PF24 COUNTER *
 23 ENTER Command here ->
 24
```

Figure 10. Sample menu showing bottom line truncated after line addition

Center lines (CE)

To center a line of text, place the cursor in the prefix area of the line to be centered and enter "ce".

Left- or right-justify lines (LE, RI)

To left-justify a line of text, place the cursor in the prefix area of the line to be left-justified, and enter "le".

To right-justify a line of text, place the cursor in the prefix area of the line to be right-justified, and enter "ri". Right-justified text may *appear* to fall off the right end of the menu. Press PF11/PF23 to scroll left and view the rightmost columns, which are normally out of view in Edit mode because of the four prefix area spaces.

Delete lines (D, DD)

Lines can be deleted two ways. Place the cursor in the prefix area of the the line to be deleted and enter a "d" to delete a single line. To delete more than one line, enter the number of lines immediately preceding the "d". For example, to delete lines 2 and 3, place the cursor in the prefix area of line 2 and enter "2d".

Another way to delete more than one line is to move the cursor to the first line to be deleted and type "dd". Then, move the cursor to the last line to be deleted, type "dd", and press ENTER. This deletes a block of lines.

Shift lines (nR, nL)

Shift commands can be used to vertically align input areas once a menu is completed.

Lines can be shifted to the left or the right. Place the cursor in the prefix area of the line to be shifted. Enter "R" to shift it one space to the right or enter "nR" to shift the line right *n* characters. To shift the line one space to the left, enter "L" or enter "nL" to shift the line left *n* characters.

Note: Characters and/or fields that shift past the right- or leftmost column of the line are truncated.

Expand lines (nE)

Blanks can be inserted into the middle of a line by following these steps:

1. Place the cursor in the prefix area of the line to be expanded.
2. Enter "E" to insert one blank.
3. Enter "nE" to insert *n* blanks.
4. Move the cursor to the position where you want the blanks inserted.
5. Press ENTER to insert the blanks.

Fields to the right of the expansion point are shifted accordingly, and any fields that are shifted past the rightmost column are lost.

You can return to Input mode to place data into the expanded area.

Shrink lines (nS)

Text or spaces can be removed from the middle of a line by following these steps:

1. Place the cursor in the prefix area of the line to be compressed.
2. Enter "S" to remove one character or space.
3. Enter "nS" to remove *n* characters or spaces.
4. Move the cursor to the position where you want the characters or spaces removed.
5. Press ENTER to remove the spaces or characters.

Fields compressed out of the line are lost.

Move lines (M, MM, B, P, F, T)

Lines are moved in two steps: first, the lines to be moved are designated; then, the location where they are to be moved is specified.

Lines to be moved can be designated in either of the following ways:

- Place the cursor in the prefix area of the line to be moved and enter "m". If more than one line is to be moved, enter the number immediately preceding the "m". For example, if two lines are to be moved, enter "2m".
- To move a block of lines, place the cursor in the prefix area of the first line to be moved and enter "mm". Place the cursor in the prefix area of the last line to be moved and enter another "mm".

Follow these steps to specify the location where the line or lines are to be moved:

- Place the cursor in the prefix area of the line after which text is to be moved and enter "f" (following) or "t" (to).
- Place the cursor in the prefix area of the line before which text is to be moved and enter "p" (preceding) or "b" (before).

Lines are actually moved—no copy remains in the original position.

Duplicate lines (", "")

You can duplicate lines in one step, in either of these two ways:

- To duplicate a line once or *n* times do the following: Place the cursor in the prefix area of the line to be duplicated and enter a double quote ("). To duplicate a line more than once, enter the number of additional copies desired immediately preceding the double quote ("). For example, to duplicate a line twice, resulting in a total of three identical lines, enter 2". The text is duplicated on the lines immediately following the original text.
- To duplicate a block of lines do the following: Place the cursor in the prefix area of the first line and enter two double quotes (""). Then, place the cursor in the prefix area of the last line to be duplicated and enter two more double quotes (""). The lines are duplicated in the area immediately following the lines being duplicated.

  Note: Duplicated lines do not require a destination. The implicit destination for duplicate lines is immediately following the duplicated lines.

If fields in the source lines were named, the duplicated lines will have the same fields and attributes but the fields will be unnamed. Lines that fall off the bottom of the menu because of duplicating are lost, as are their field names and attributes.

Copy lines (C, CC, B, P, F, T)

Lines are copied to other locations in the menu in two steps: first, the lines to be copied are marked; then, the area where the lines are to be copied is specified.

Lines to be copied can be marked in the following ways:

- Place the cursor in the prefix area of the first line to be copied and enter "c". If more than one line is to be copied, enter the number of lines to be copied immediately preceding the "c". For example, if three lines are to be copied, enter "3c".
- To copy a block of lines, place the cursor in the prefix area of the first line to be copied and enter "cc". Then, move the cursor to the prefix area of the last line to be copied and enter another "cc".

The area where the marked line or lines are to be copied is specified by one of the following methods:

- Place the cursor in the prefix area of the line **after** which text is to be moved and enter "f" (following) or "t" (to).
- Place the cursor in the prefix area of the line **before** which text is to be moved and enter "p" (preceding) or "b" (before).

If fields in the source lines were named, the copied lines will have the same fields and attributes but the fields will be unnamed. Lines that fall off the bottom of the menu because of copying are lost, as are their field names and attributes.

Import lines from a CMS file (G, GG)

You can incorporate text from other files into XMENU menus using the G (for "get") and GG prefix subcommand. However, you **cannot** incorporate lines from another XMENU menu; these menus are stored in binary format.

If your menu becomes longer than specified as a result of importing lines from a CMS file, the excess lines will be truncated from the bottom.

Figure 11 on page 35 shows a sample menu with the G command in the prefix area. Note that the GG block sets aside lines 12 through 15 for the imported text; that is, four lines. The same result could be obtained by typing 4G on line 11.

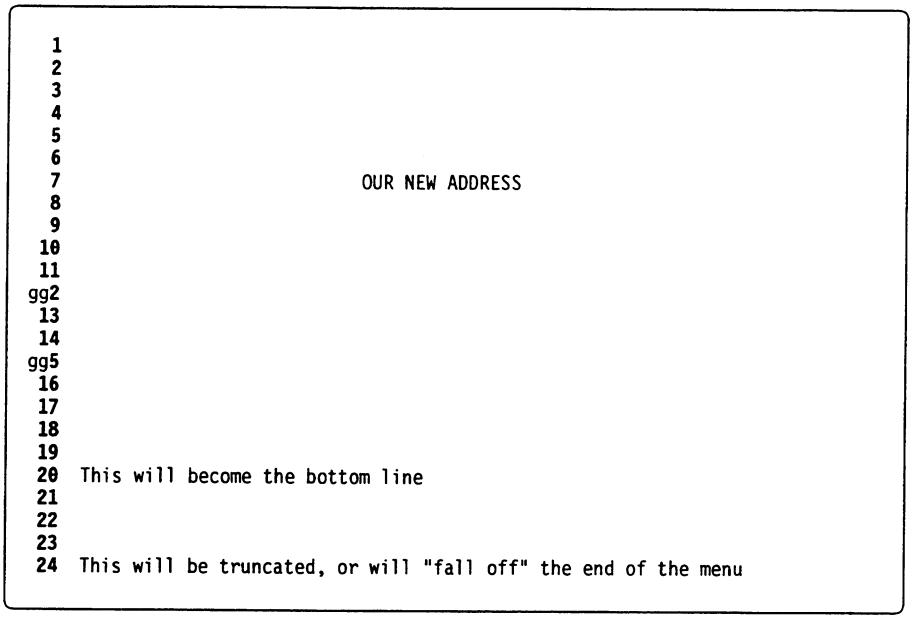


Figure 11. Sample edit command to import CMS file lines

After pressing ENTER, the screen shown in Figure 12 on page 36 appears, on which you specify the remaining information about your CMS file:

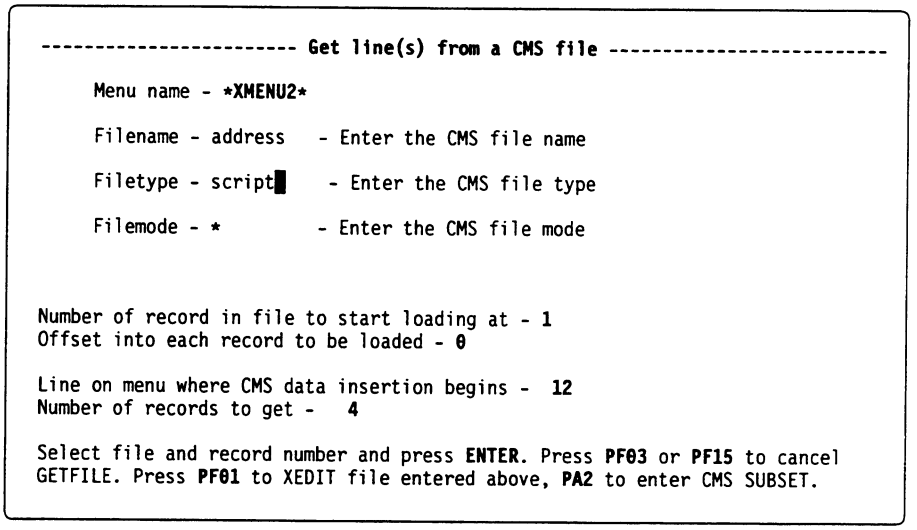


Figure 12. XMEDIT screen for importing CMS file lines

You have a few options when using the import menu:

- Supply the appropriate entries and press ENTER.
- Press PF3/PF15 to return to Edit mode without importing any lines.
- Press PF1/PF13 to XEDIT the specified CMS file.

The import menu has some default settings that you can accept or type over:

Fields for importing CMS file lines

`Menu name`	XMENU has put the default menu name *XMENU2* here. You cannot change this; XMENU "knows" the name of the menu you are working on.
`Filename`	Specify the filename of the CMS file you want to import.
`Filetype`	Specify the filetype of the CMS file you want to import.
`Filemode`	XMENU has put the filemode * (any) here. There is no need to specify the filemode of a uniquely named CMS file that you want to import. If more than one file has the same name and type as the one you desire, enter the appropriate filemode to select the correct file.
`Number of record in file to start loading at`	Specify the line number of first line to be imported. The default is line 1.
`Offset into each record to be loaded`	If the needed data begins farther to the right than the first column, specify how many columns to the right it does begin. No offset at all is 0 (column 1), the default.
`Line on menu where CMS data insertion begins`	XMENU has filled this in from your specific invocation of the `G` command in the prefix area. You cannot change the number in this field from this screen; you must exit to Edit mode and re-enter the `G` command.
`Number of records to get`	XMENU has filled this in from your specifications in the prefix area. You cannot change the number in this field from this screen; you must exit to Edit mode and re-enter the `G` command.

To complete our example, once the fields in the import menu are entered or changed as needed, and ENTER is pressed, you are returned to Edit mode, with the specified lines placed accordingly:

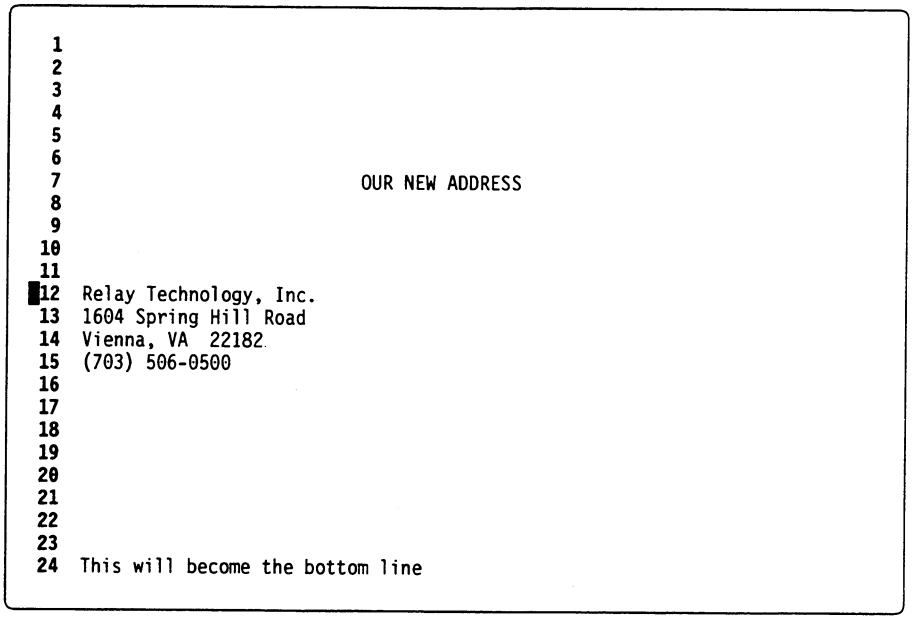


Figure 13. Sample edit screen after importing lines from a CMS file

Completing the menu from Input mode

After you enter and position the text for your menu to your satisfaction, press PF3/PF15 to exit from Edit mode and return to Input mode.

Here, you will concentrate on assigning field attributes, naming the fields, and specifying the position where the cursor is to appear when the menu is first displayed.

PF keys are used to perform each of these functions. Once assigned in Normal Input mode, special characters are displayed on the input screen to mark the fields and to indicate which fields are named. We will assume Normal Input mode is being used in this chapter; details on the differences between Normal and Alternate Input mode are presented in "Alternate Input mode" on page 19.

When you define a field by assigning field attributes, you actually specify several things about the field at one time:

- You define the location of the field on the menu—where it starts, its length, and where it ends.
- You set the attributes of the field—how data in the field will appear to the user when it is displayed or typed in the field.

When you name a field, you provide an application or program the ability to refer to the field to do one or both of these two things:

- Accept and use data entered by a user in a field
- Display data in a field

PF key settings for Input mode

Program Function keys have different purposes in Input mode. They are used for a variety of tasks, from assigning field attributes, to naming fields and displaying the results on the screen. The following chart describes these default settings for PF keys in Input mode, as well as for PA1, PA2, CLEAR, and ENTER:

PF keys for Input mode

Key	Purpose
PF1	Displays a pop-up command menu on which you can enter CMS and CP commands as well as XMEDIT subcommands (see Figure 26 on page 50).
PF13	Displays the default PF key definitions for Input mode.
PF2/PF14	Exits from XMEDIT ***without*** saving changes you might have made to the menu during this XMEDIT session. XMEDIT prompts you to confirm that you really want to exit without saving changes.
PF3/PF15	Saves the menu on your A-disk and exits from XMEDIT.
PF4/PF16	Defines the start of a protected, bright field at the cursor position.
PF5/PF17	Defines the start of a protected, normal (dim) field at the cursor position.
PF6/PF18	If the cursor is on a field marker at the beginning of a field, this key displays an action bar where you can select or modify the field attributes. The action bar shows extended attributes if applicable. If the cursor is not on a field marker, this key displays the action bar, as above. If you simply press PF3/PF15 to quit the action bar, you define the start or end of a field with the default attributes: • Unprotected • Normal (dim) display • SKIP not set • MDT flags not set • If extended attributes are being used, default highlighting, color, and programmed symbol set are also assigned
PF7/PF19	Defines the start of an unprotected, bright field at the cursor position.
PF8/PF20	Defines the start of an unprotected, normal (dim) field at the cursor position.
PF9/PF21	Defines the start of an unprotected, nondisplay (dark) field at the cursor position.
PF10/PF22	Changes all characters to uppercase from the cursor position to the next blank or non-alphabetic character.
PF11/PF23	Records the current position of the cursor, to set its initial display position on your menu. If this isn't specified, the initial cursor position is in the upper-left corner of the screen.
PF12/PF24	If the cursor is on a field marker at the beginning of a field, displays the field naming screen and moves the cursor one position to the right after a name is entered. If the cursor is not on a field marker, moves the cursor to the next field marker.
PA1	Transfers to Display mode, showing the menu you are creating as it will appear to the user. Return to Input mode with any interrupt-generating key except PA2.
PA2	If the cursor is positioned on a field marker, erases the field marker and removes the field. If the cursor is not on a field marker, it is moved backward to the beginning of the preceding field.
CLEAR	Has no XMEDIT function. Any input entered before pressing CLEAR is lost.
ENTER	Transfers to Edit mode.

Notes on working in Input mode

This section contains more details on working in Input mode to define field attributes, name fields, and set the initial cursor position. You may want to skip this section if you feel comfortable performing these tasks from the PF-key driven Input mode environment. To complete your understanding of the XMEDIT environment, continue with "Using Display mode" on page 47.

Define fields by assigning attributes

In this section, we step through the process of defining fields by assigning attributes to one field in the sample menu below.

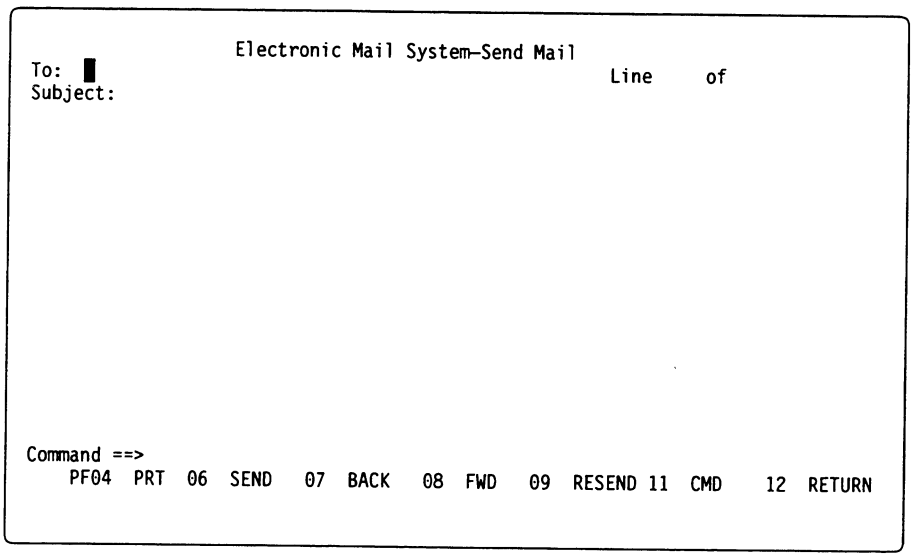


Figure 14. Sample menu ready for field assignments

Using the attribute action bar and pull-down windows: You assign field attributes by positioning the cursor where you want the field to begin and pressing PF6/PF18. An action bar appears on the top of your screen. You can enter one- or two-character commands in the input area to the left of this bar. The optional second character specifies the value for the attribute type indicated by the first character. If you enter only one character, a pull-down menu appears that gives you the valid value choices for the chosen attribute type. The correct single letter code is the letter that is capitalized in each word on the action bar and in the pull down menus; usually it is the first letter of the word.

Follow these three steps to define the start of a field:

1. Display the menu on the input screen.
2. Position the cursor where you want the field to begin.
3. Press PF6/PF18 to display a pop-up action bar, where you can specify the desired attributes.

On the sample electronic mail menu, the area after **To:** is where the user will enter the userid of the person being sent mail. Since this field will accept user input, it should be unprotected, and because we want to make sure the user enters a userid, we will

highlight the input data by giving the field the bright attribute. We'll also assign the color yellow to this field to heighten its visual impact for users with color terminals that support extended attributes (Refer to "Extended attributes" on page 15 and Appendix C, "Attribute types and values" on page 121, for a discussion and listing of the basic and extended attributes that that you can assign to fields).

To assign the field unprotected, bright, and yellow attributes follow these steps:

1. Place the cursor two spaces after **To:**.
2. Press PF6/PF18.

   A pop-up action bar is displayed where you can set attributes. If your menu was created with no extended attributes, the attribute action bar looks like this:

```
 Protect  Intensity  Skip  MDT  Quit
```

Figure 15. XMEDIT attribute selection pop-up action bar

With extended attributes, the attribute action bar looks like this:

```
 Protect  Intensity  Skip  Mdt  Color  Highlight  Logical-symbol-set  Quit
```

Figure 16. XMEDIT extended attribute selection pop-up action bar

3. Type "p" for Protect.
4. Press ENTER. A pull-down window appears where you can set the protection you want for the field. The current setting is shown:

```
p     Protect  Intensity  Skip  MDT  Quit
  U   Protected
      Unprotected
      Numeric
      Quit
```

Figure 17. XMEDIT pull-down window of field protection choices

The current value, "U" for Unprotected, is already displayed on the pull-down window. Notice that if you wanted to assign either Protected or Numeric protection to this field, you could type "p" or "n" as your choice.

5. Press ENTER. The pull-down window disappears and the field is now unprotected.
6. Follow similar steps to set the other attributes.
   - Type "i" for Intensity.
   - Press ENTER. This pull-down window appears:

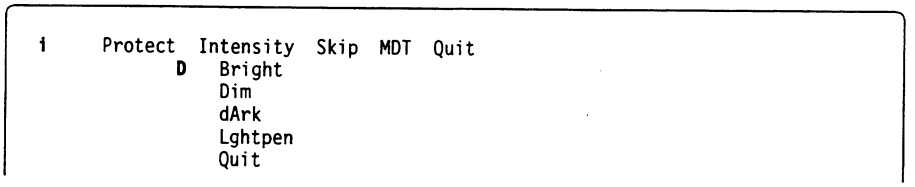


Figure 18. XMEDIT pull-down window of field intensity choices

- Type "b" for Bright. Notice that to assign a field the Dark attribute, you would type "a".
- Press ENTER.

  The field is now unprotected and bright.

We will assign only one other field attribute—color, an extended attribute—in this discussion. For a complete list of basic and extended field attributes that will appear on these windows, refer to Appendix C, "Attribute types and values" on page 121.

Once you become familiar with the field attribute values available on the pull-down windows, you can enter two characters directly into the action bar field. For example, to make this field yellow, you could enter a "c" and press ENTER to get the color choice pull-down window shown below, and type "y" for yellow and press ENTER, or you can simply type "cy" directly in the action bar input area and press ENTER.

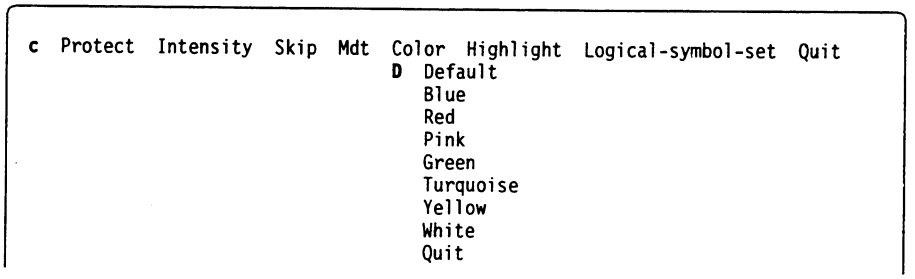


Figure 19. XMEDIT extended attribute selection action bar and pull-down window of color choices

Once you have assigned the attributes you desire, press PF3/PF15 or "q" for quit, to exit the action bar pop-up window.

Returning to Input mode from attribute selection: After leaving the attribute selection window, you are once again in Input mode where a split vertical bar now appears at the start of the input field following **To:**. The split vertical bar marks the beginning of the bright, unprotected, yellow field. The menu now looks like this:

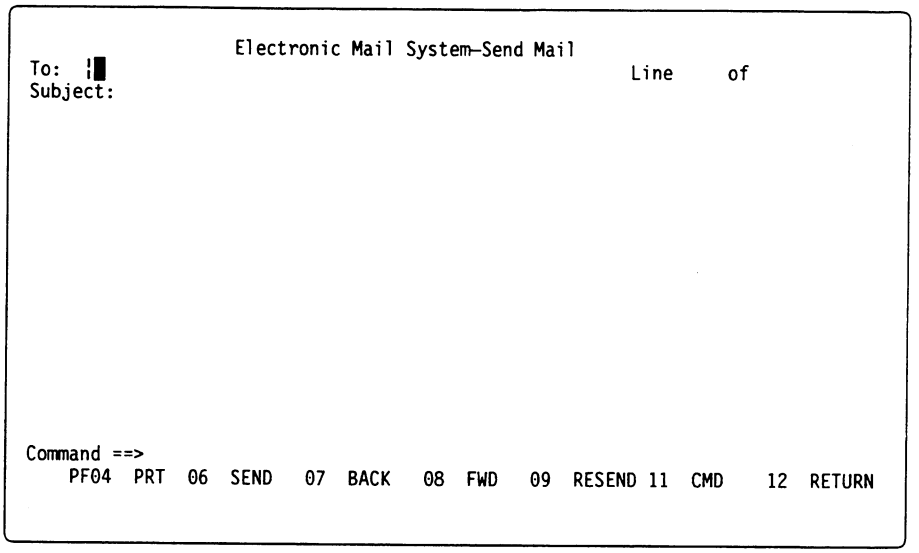


Figure 20. Sample menu showing field's first attribute marker in position

The split vertical bar is the attribute character or field marker; it indicates the starting point of a specified attribute or attribute combination. This is the default attribute character. If necessary, you can change the character used. For more information, see "Field-marking characters" on page 21.

Note: Do not type a split vertical bar when you are defining attributes. XMEDIT does not recognize the typed split vertical bar as an attribute character and treats it just like any other typed character.

Marking the end of the field: The field you just assigned yellow, unprotected, and bright attributes has no end. The attributes are in effect from the just-created field marker to the next field marker, which, in this case, means that the whole menu will display as an unprotected, bright, yellow field. Attributes extend across the line, down and across the next line, and so on past the bottom right corner of the screen to the upper left corner of the screen, and so on. Our job now is to end the field by signalling the beginning of another.

To mark the end of the **To:** field, we will allow eight spaces for users to enter a userid.

1. Space over eight spaces, positioning the cursor on the ninth space after the beginning field marker.
2. Press PF6/PF18.

   The pop-up attribute selection window is displayed.
3. Assign the protected and dim attributes and either press PF3 or type "q" to quit the window and save your choices.

A second split vertical bar marks the end of the **To:** field. It also marks the beginning of a protected, normal intensity (**DIM**) area that is in effect from the end of the **To:** field to the beginning position of the next field.

The menu now looks like this:

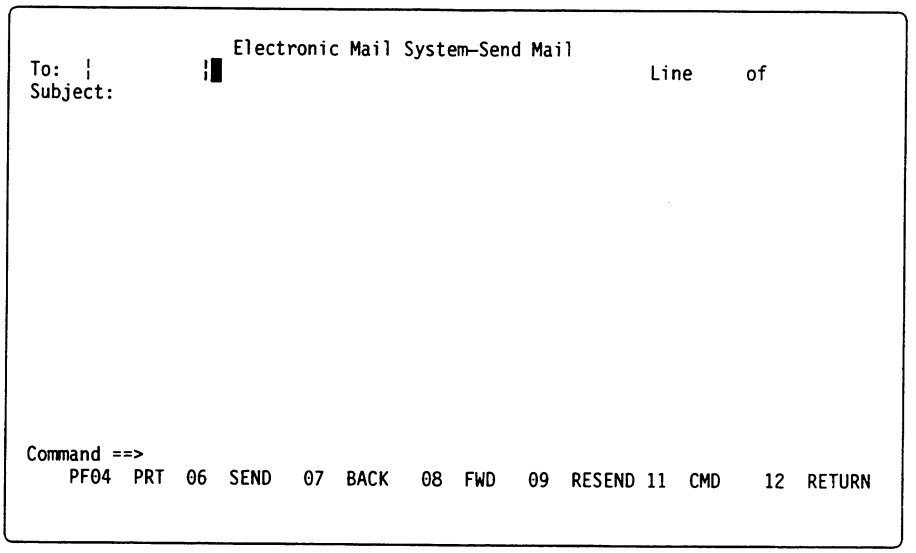


Figure 21. Sample menu showing field's start and end attributes assigned

Review and modify field attributes

To review or modify any of the field attributes you have set, simply place the cursor on the split vertical bar that marks the field whose attributes you want to review or modify and press PF6/PF18. The pop-up action bar will appear, as described above. You can review the attribute settings and change any of them by typing over the attribute command letter with your new choice. Then, either press PF3/PF15 or enter "q" to quit to save the new attribute. You can quit from the display without changing an attribute by either pressing PF3/PF15 or entering "q" to quit.

Delete a field

You delete fields differently when in Normal or Alternate Input mode. For more information on these two modes, refer to "Alternate Input mode" on page 19. You must be careful about how you delete a field; relocating a field marker character destroys the field.

To delete a field while in Normal Input mode, place the cursor on the character that marks the field you want to delete and use the space bar to delete the field marker character. The field is now deleted.

To delete a field while in Alternate Input mode, place the cursor at the start of the field you want to delete and press PA2, the key defined to perform the DESTROY function in Input mode. The field is now deleted.

Name a field

Because an application must be able to refer to a field if the field is used for data entry or display, any field that an application references must have a name. XMENU menu fields are named so that the application program(s) developed to support your menu(s) can reference fields by symbolic name rather than by menu offsets or field numbers. Therefore, if you decide at any time to move a field to a different position on the menu, it is possible to do so without making any changes to your application.

Field names must begin with an alphabetic character, have from one to seven characters, and be unique. Numbers, alphabetic characters and the following symbols can appear after the first character of the field name: period (.), dollar sign ($), at sign (@), number sign (#), exclamation mark (!), question mark (?), cent sign (¢), and underscore(_).

To name a field using the Name-a-field pop-up window, follow these steps:

1. Place the cursor over the field marker character—the split vertical bar—that marks the beginning of the field you want to name
2. Press PF12/PF24 to display the Name-a-field pop-up window. See Figure 22.
3. Type a valid field name in the space provided.
4. Press ENTER.

The small Name-a-field window overlays the top right portion of the menu on which you are working until you press ENTER to assign or remove a field name, or you press PF3/PF15 to quit:

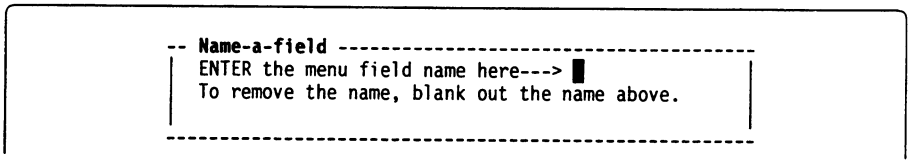


Figure 22. Name-a-field pop-up window

After you press ENTER from the field naming window, the split vertical bar attribute character that appeared at the beginning of the input field is replaced by a tilde (~), indicating that the field has a name:

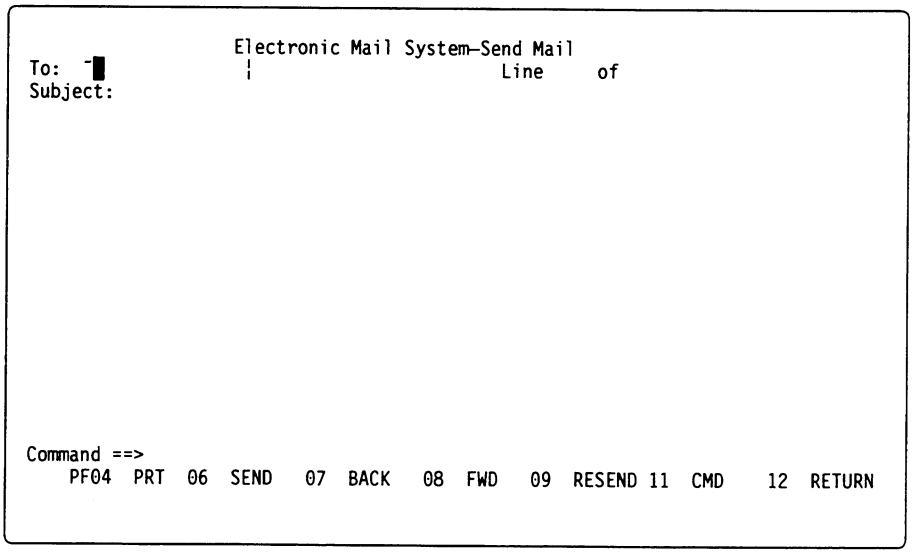


Figure 23. Sample menu showing a named field

Set the initial cursor position

You can specify where the cursor is to be positioned when your menu is first displayed by positioning the cursor in the desired location and pressing PF11/PF23 while in Input mode. If you did not specify an initial cursor location, the cursor appears in the upper left corner of the menu.

The screen is simply refreshed when the initial cursor position is defined; no special field markers are used.

Normally, you will want the cursor to appear on the first input position of the first unprotected field on the menu, where users will probably first enter data. This helps users so they don't waste time moving the cursor around the menu.

Using Display mode

Display mode presents your menu as it will be seen and used by the users. From Display mode you can also print the menu.

Enter Display mode by pressing the PA1 or TEST REQUEST (SYS REQ) key from the Input mode screen. To return from Display mode to Input mode, press PF3/PF15, ENTER, or any other interrupt-generating key except PA2. (PA2 in Display mode is set to create a print file of your menu.)

Menus seen in Display mode will exhibit all the characteristics and attributes you have assigned, limited only by the type of terminal you are using. For example, if you assigned a color to a field, its color display requires a terminal capable of supporting extended attributes. Viewed on a terminal without extended attribute capabilities, this field will display without error, but also without color. Field protection, intensity, highlighting, etc. are in effect when you view the menu in Display mode.

The figure that follows shows how the Input mode example menu would appear in Display mode. This figure shows the menu after a sample userid is entered in the field whose attributes were defined in the previous section:

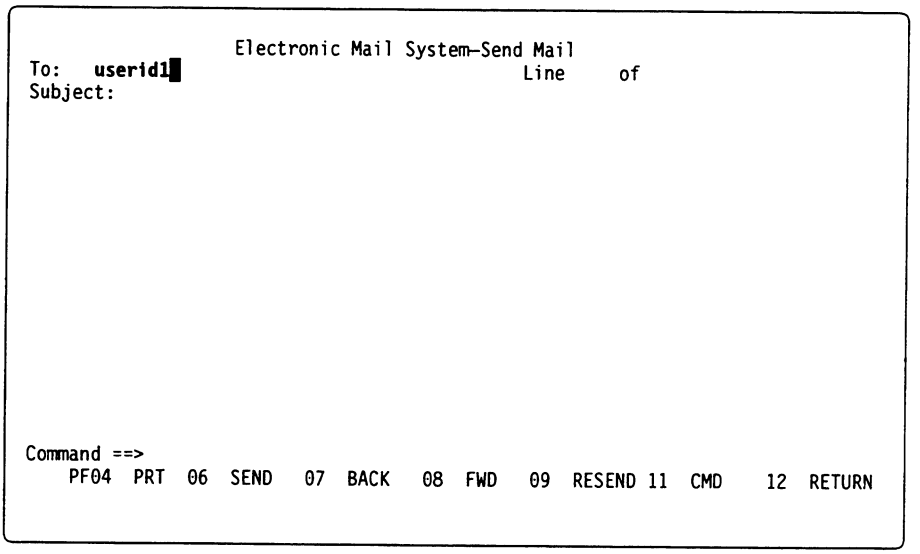


Figure 24. Sample menu shown in Display mode

Keys active in Display mode

The following keys are active while in Display mode:

Keys available in Display mode

Key	Purpose
PA2	Prints the menu and sends a print file to your virtual printer. If you specified either the SCRIPT or DISK option, a ***menuname*** **SCRIPT** file or a ***menuname*** **LISTING** file, respectively, is placed on your A-disk.
Any other key *except* PA2	Transfers back to Input mode.

Notes about Display mode

You can alternate between Input and Display modes at any time to ensure that the menu layout and field definitions are progressing as desired, the cursor is positioned where you expect it, unprotected fields allow data entry, highlighted fields are displayed as highlighted, and so on. From Display mode you can also create a print file of your menu that also includes a detailed report on field locations, attributes, field names, etc.

Test the fields with data in Display mode

In Display mode, a menu looks and functions exactly as it will when users are using it. You can even type data into fields while in Display mode. Although your input will be ignored, you will be able to see how the field is accepting data and highlighting it, depending on the attributes you assigned. While you are creating a menu it is a good idea to check every once in a while to see what the menu looks like and how it functions.

You should use the Display mode screen to test your menu for the following conditions:

- Is the text properly positioned and the overall look of the menu balanced and clean?
- Are the areas that simply display text protected so that you cannot type data into those fields?
- Are all the input fields unprotected so you can type data into them?
- Is the position of the cursor when the menu is first displayed where you expect it to be? If you did not specify an initial cursor location, the cursor appears in the upper left corner of the menu.
- Are the appropriate fields highlighted?
- If you set the SKIP option, does the cursor automatically skip over protected fields?

Print the menu

After you create a menu, you might find it helpful to have a printed copy of it. An XMEDIT print file includes a "screen print" of the menu as well as a list of all the menu's fields, their names, attributes, lengths and screen positions.

To print a copy of the menu, press PA2 in Display mode. A printed copy of the menu is sent by default to your virtual printer. Each time you print the menu a new copy is appended to the open print file in your virtual printer. The print file is closed when you leave XMEDIT.

XMEDIT produces printed output for printers that can recognize ANSI (ASA) control characters in position one of the record, such as IBM 1403 printers. If you do not have a 1403 printer, you must reroute the printed output to a disk file so you can edit it and print it on your device. You can also reroute the print output to create a file in DCF (SCRIPT/VS) format. For more information on creating LISTING and SCRIPT files, see "Send print output to a disk file" on page 20.

It is possible to print the menu without entering Display mode by using the PRINT subcommand. For more information on issuing XMEDIT subcommands from Input mode see "The XMEDIT command line" on page 50. Refer to Part 2, The XMENU editor utilities reference, "PRINT subcommand" on page 87 for details on the PRINT subcommand. This subcommand can be assigned to a PF key, if you desire. Refer to "Modify XMEDIT's default PF key settings" on page 25 for more details.

To return to XMEDIT Input mode, press any interrupt-generating key except PA2.

Saving the menu and/or discarding menu changes

To leave XMEDIT and file the menu for use or further work, press PF3/PF15 from Input mode. The menu is saved on your A-disk with the filename you specified and the filetype MENU.

If you want to quit working on the menu without saving a copy, press PF2/PF14 from Input mode. The following screen will be presented asking you to confirm that you want to quit:

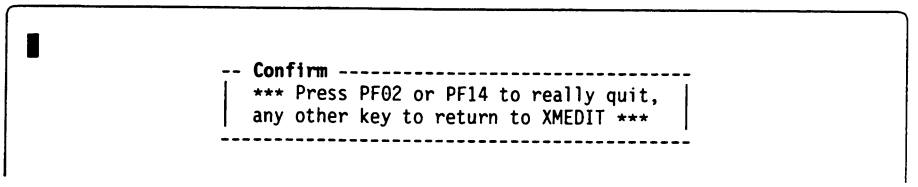


Figure 25. XMEDIT quit confirmation screen

Press PF2/PF14 again to quit from XMEDIT. No file is saved on your A-disk.

If you have previously saved a copy of the menu, and want to re-enter XMEDIT to make additions or changes to the menu, enter the XMEDIT command with your menu name:

```
XMEDIT menuname
```

You will enter Input mode where your menu appears with all the previously set options in effect.

If you want to quit from this XMEDIT session without saving changes, press PF2/PF14 from Input mode, as above. Any changes that you may have made **during the most recent XMEDIT session** are not saved. The menu will exist on disk as it was before.

The XMEDIT command line

There are many advanced functions available in the XMEDIT program that you do not need when you are first learning to develop menus.

XMEDIT has its own set of subcommands, many of which simulate user input into XMEDIT Input mode, and many of which are based on, or modify the position of, XMEDIT's Input and Edit mode cursor.

XMEDIT subcommands are most often used in XMEDIT macros, which are discussed in more detail in Part 2, The XMENU editor utilities reference, "XMEDIT macro facility" on page 62. However, these subcommands can also be issued from XMEDIT Input mode.

In Input mode, press PF1 (**not** PF13) to display the following pop-up command line menu:

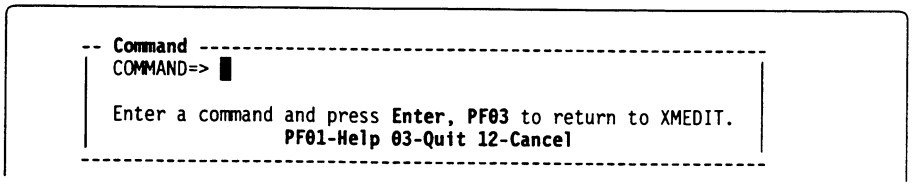


Figure 26. XMEDIT pop-up command line

This command line can be used to issue CP and CMS commands as well as XMEDIT subcommands.

A list of some of the more commonly used subcommands is included in Appendix F, "Changing the defaults by editing PROFILE XMEDIT" on page 127. Later, as your familiarity with the product grows and as your desire for more advanced XMENU applications develops, you will want to refer to Part 2, The XMENU editor utilities reference, "XMEDIT macro facility" on page 62 and "XMEDIT subcommands" on page 62 where all the XMEDIT subcommands and their calling environments are described.

Part 2. The XMENU editor utilities reference

This section provides a complete reference for XMEDIT and its related utilities. Here's a "road map" of the utilities presented in this reference section:

Chapter 3. XMEDIT

The XMENU menu editing program, XMEDIT, is used to create and update menus used by the XMENU/REXX Interface (MENUEXEC) and the XMENU High-Level Language subroutines. This utility allows you to interactively create and modify menus, change field attributes, and name fields. You can also use it to move a field from one place to another, to create a new menu using an existing menu, and to incorporate portions of existing data files into a menu.

XMEDIT command format

To invoke the XMEDIT editor, use the following command format:

XMEDIT XMENU	[*menuname*|?] [(*options*][)] [*menuname*|?] [(*options*][)]

Where

menuname Specifies the name of the XMENU menu to be created or modified. If a menu with this name was loaded into storage using XMENUINS, it will be used, even if the option LIB is specified. (XMENUINS is documented in the *XMENU Subroutine Library Reference* manual.)

? Displays the XMEDIT editor HELP menu.

options Specifies one or more XMEDIT options. See "XMEDIT options" on page 56 for a detailed description of the options.

If the XMEDIT command is entered alone, that is, without a menu name or options, a prompting screen, the XMEDIT Options Screen, is displayed to allow you to enter the name of the menu and choose XMEDIT options.

If you use the alternate command, XMENU, the OLDWAY option is implicitly set. This provides editor screen format compatibility with previous versions of the XMENU product. See Appendix D, "Assigning attributes using the OLDWAY full-screen menus" on page 123 and Appendix E, "Naming fields by using the OLDWAY full-screen menus" on page 125 for more information.

XMEDIT options

All the XMEDIT options are presented in the alphabetical listing below:

Option	Action
ALARM	The terminal alarm is sounded whenever the menu is displayed. This option may be overridden by the calling EXEC or high-level language application.
ALT	Alternate Input mode will be used rather than Normal Input mode. Alternate Input mode uses actual attributes rather than characters to mark the positions of menu field attributes. ALT mode editing may be set in the XMEDIT PROFILE or changed during an edit session by using the `SET ALT OFF|ON` subcommand.
AUTONAME	Program-assigned names will be given to each field on the menu after the menu is edited. This option overwrites existing field names. The names are assigned to fields in screen position order, normally in the form *NNNNxxx*, where: *NNNN* — The first four characters of the menu name. If the menu name is less than five characters the entire menu name is used. *xxx* — A three-digit number assigned in ascending order starting with 001. Certain other XMEDIT options may alter the name assignment format. The options TRUNC, FNAME, FNUMB and FGROUP will be ignored if AUTONAME is not also specified as an XMEDIT option.
CHAR *char1 char2*	Specifies the two special characters used to show the positions of menu fields. *Char1* is the character displayed for unnamed fields. *Char2* is the character displayed for named fields. Both should be either specified as two-digit hexadecimal numbers, or a single character. If CHAR is not specified, the default character for *char1* is X'6A'; the default for *char2* is X'A1'. The choice of characters must be between X'41' and X'FE'. The CHAR characters may be set in the XMEDIT PROFILE or changed during an edit session by using the `SET CHAR` subcommand.
DISK	Printed output will go to a CMS file named *menuname* LISTING. If DISK is not specified, printed output goes to the virtual printer.
DSECT	Following the creation or modification of the menu, an assembler DSECT of the menu is created.
EXT	Allows the definition of extended attributes within your menu (that is, extended highlighting, color, and programmed symbol sets). If you are modifying an old menu without extended attributes, it is converted to extended attribute format and existing

fields are assigned default extended attributes. If you are using the XMEDIT Options Screen, extended attributes are automatically selected if you are running on a terminal supporting them. Menus with and without extended attributes can be used on terminals with or without extended attribute capabilities. If you have no extended facility terminals, there is some performance gain in only creating menus without extended attributes.

NOEXT specifies that XMEDIT create this menu without extended attributes.

Extended attribute usage may be set in the XMEDIT PROFILE using the `SET EXT OFF|ON` subcommand.

FGROUP *start-name end-name*
Specifies that automatic field naming be performed for only the fields between and including the existing fields named *start-name* and *end-name*. If only *start-name* is specified, every field from and including that named field to the end of the menu is AUTONAMEd.

FNAME *name*
Specifies that, when assigning field names via AUTONAME, the name specified in FNAME be used rather than the menu filename. One to four characters can be specified. FNAME can be used to automatically assign the same names to groups of fields in different menus.

FNUMB *number*
Specifies that, when assigning field names via AUTONAME, the starting number specified in FNUMB be used rather than starting with 001.

LIB *libname*
Specifies the library name from which the menu is to be loaded for editing. LIBNAME refers to the filename of an XMENULIB; however, the updated menu is written to a discrete CMS file with filetype MENU. If the source menu was previously loaded into storage by XMENUINS, this option is ignored.

MDT
Sets the Modified Data Tag (MDT) on newly created unprotected fields. This option only sets the MDT on fields created or modified during this XMEDIT session. Previously created fields not modified in this session are not changed.

Automatic setting of MDT bits may be set in the XMEDIT PROFILE or changed during an edit session by using the `SET MDT OFF|ON` subcommand.

MENTEXT
Creates an object file and an assembler DSECT of the menu following the creation or modification of the menu.

NAME
Once the menu has been created, the XMEDIT editor prompts you to name each field in the menu. Whether this option is used or not, fields may be named individually during menu creation using the Name-a-Field Screen.

NAMEUNP
Only unprotected fields are automatically named. The NAMEUNP option implicitly sets AUTONAME. NAMEUNP used with the REXX option will only name unprotected fields with REXX-style names.

NOALARM The alarm is not sounded each time the menu is displayed. This option is the default option. NOALARM can be used to clear the specification of ALARM in a previous XMEDIT editing session.

NOALT Normal Input mode is to be used. This is the default.

ALT mode editing may be set in the XMEDIT PROFILE or changed during an edit session by using the `SET ALT OFF|ON` subcommand.

NOEXT The menu will be created without extended attributes that is, extended highlighting, color, and programmed symbol sets. If you are using the XMEDIT Options Screen, NOEXT is automatically selected if you are running on a terminal that does not support extended attributes. Menus with and without extended attributes can be used on terminals with or without extended attribute capabilities. If you have no terminals capeable of displaying extended attributes, there is some performance gain in only creating menus without extended attributes.

Extended attribute usage may be set in the XMEDIT PROFILE using the `SET EXT OFF|ON` subcommand.

NONEWM If a new menu is being edited, the Welcome Screen is not displayed before editing begins (see Figure 6 on page 26). This option can be used in conjunction with XMEDIT macros to provide batch-mode editing that doesn't force the user to respond to a confirmation screen.

NOPROF No XMEDIT macro is invoked before editing begins as an XMEDIT profile. If neither this option nor PROFILE *name* are specified, PROFILE XMEDIT is executed, if it exists.

NULLCHAR *char* Specifies a character to be used to display null (binary zero) characters in the menu. *Char* must be either a single character, or a two-digit hexadecimal number between X'41' and X'FE'.

If not specified, and no nulls exist in the menu being edited, no NULLCHAR exists, otherwise an implicit NULLCHAR of ` (X'79') is defined.

Automatic use of the NULLCHAR may be set in the XMEDIT PROFILE or changed during an edit session by using the `SET NULLchar` *`char`* subcommand.

OLDWAY Specifies that full-screen menus are used to prompt for attribute values and field names, rather than pop-up windows (see Appendix D, "Assigning attributes using the OLDWAY full-screen menus" on page 123, and Appendix E, "Naming fields by using the OLDWAY full-screen menus" on page 125). This option provides compatibility with older versions of XMENU.

Automatic use of the OLDWAY option may be set in the XMEDIT PROFILE or changed during an edit session by using the `SET OLDWAY OFF|ON` subcommand.

PROFILE *name* Specifies that a specific XMEDIT macro be invoked as an XMEDIT profile (that is, before editing begins). *Name* is the name of the XMEDIT macro file. If neither this option nor

NOPROF are specified, macro file PROFILE XMEDIT is executed, if it exists.

REXX Menu fields are automatically named with names of the form *XXX.yyy* where *XXX* is a portion of the menu name and *yyy* is a leading-zero-suppressed ascending number. There is a period (.) between *XXX* and *yyy* to allow the use of REXX variable structures. Use of the REXX option implicitly sets TRUNC and AUTONAME.

SCRIPT Printed output is written to a CMS file named *menuname* SCRIPT in SCRIPT format rather than being sent to the default virtual printer.

SIZE *xxx* <*yyy*> Specifies the size of the menu to be created, where *xxx* specifies the number of lines the menu will contain, and *yyy* specifies the number of columns the menu will contain. If *yyy* is not specified, it defaults to the terminal column size for the number of lines specified in *xxx*. The maximum value allowed for either *xxx* or *yyy* is 256.

If no size is specified, the size defaults to the largest size of the terminal on which the menu is being created. If the menu file already exists, the size of the menu overrides any value specified by the SIZE option.

When both the SIZE and USING options are used together, the vertical size (number of rows) of the menu is either lengthened or truncated to match SIZE before being displayed, while the column size of the old menu must match the column size of the new menu.

Unless you are generating menus for use in XMENU windows, the SIZE option should only be used to create menus the same size as the terminals the menus will most often be displayed on, for example, 24 by 80.

SKIP Sets the SKIP attribute on newly-created or changed protected fields. This option sets the skip attribute only on fields created or modified during this XMEDIT session. Previously created fields not modified in this session are not changed.

Automatic use of SKIP may be set in the XMEDIT PROFILE or changed during an edit session by using the `SET SKIP OFF|ON` subcommand.

TEXT Following the creation or modification of the menu, an object file containing the contents of the menu is created.

TRUNC The numbers assigned via AUTONAME are leading-zero suppressed. This eases the use of AUTONAME names in EXECs using indexed EXEC variables.

ULIB *libname* Specifies the library name from which the menu specified in USING is loaded for editing. *Libname* refers to the filename of an XMENULIB; however, the updated menu is written to a discrete CMS file with filetype MENU. If the source menu was previously loaded into storage by XMENUINS, this option is ignored.

UPCASE The entire menu will be converted to uppercase before being saved to disk.

Automatic setting of UPCASE may be set in the XMEDIT PROFILE or changed during an edit session by using the SET UPCASE OFF|ON subcommand.

USING *filename* A new menu is to be created using *filename* MENU as initial input. This option is similar to the XEDIT GET subcommand. This option is only valid when creating a new menu.

If SIZE is not specified, the default size of the new menu is the size of the terminal.

When both the SIZE and USING options are used together, the vertical size (number of rows) of the menu is either lengthened or truncated to match SIZE before being displayed, while the column size of the old menu must match the column size of the new menu or an error message will be issued.

XMEDIT error messages and return codes

The following list provides, in numeric order, the XMEDIT return codes and error messages.

Return codes and messages

8100 8100E Too many parameters specified.
8101 8101E Invalid option specified.
8102 8102E Invalid or missing CHAR option characters.
8103 8103E Old menu column size is larger than terminal size.
8104 8104E Cannot edit a menu with more than 256 columns/line.
8105 8105E LIB option filename is missing or invalid.
8106 8106E Insufficient storage to run XMENU.
8107 8107E ULIB option filename is missing or invalid.
8108 8108E USING filename is missing or invalid.
8109 8109E Cannot specify LIB and ULIB together.
8110 8110E SIZE option missing, invalid or bigger than 256.
8111 8111E Terminal must be at least 20 x 80 to run XMENU.
8112 8112E Terminal screen size is not supported.
8113 8113E Terminal I/O error occurred.
8114 8114E Menu or USING filename has invalid characters.
8115 8115E USING menu file or XMENULIB member not found.
8116 8116E XMENULIB library not found.
xxxx 8117E Error reading the menu or USING file.
8118 8118E Menu must not already exist if using the USING option.
8119 8119E Menu or USING file is not proper XMENU format.
8120 8120E XMENU library specified is not in XMENULIB format.
8121 8121E XMENU library member requested does not exist.
8122 8122E Menu format error. SFA W/O corresponding field block.
8123 8123E Old menu line/column size is larger than terminal size.
8124 8124E FGROUP beginning or ending field names missing.
8125 8125E FNUM number missing or non-numeric.
8126 8126E FNAME field name is missing.
8127 8127E FNUM number is not between 1 and 9000.
8128 8128E No R/W disk - access one as "A" then enter "RETURN."
8129 8129E Disk full - free disk space or re-access another disk as "A" and enter "RETURN."
xxxx 8130E Error writing MENU file. Check FSWRITE return code.
0 8131E Autoname prefix characters invalid. AUTONAME ignored.
0 8132E FNUM truncated value too large for AUTONAME. AUTONAME terminated.
8133 8133E Automatic field naming terminated due to errors.
8134 8134E Insufficient storage to automatically name fields.
8135 8135E Duplicate field name encountered naming fields.
xxxx 8136E Error in state of object (TEXT) file.
xxxx 8137E Error writing object (TEXT) file.
xxxx 8138E Error writing copy (DSECT) file.
8170 8170E Unable to define virtual screen.
8171 8171E Unable to define window.
8172 8172E PROFILE filename is missing or invalid.
8173 8173E The NULLCHAR character is missing or invalid.

XMEDIT macro facility

The XMEDIT menu editor supports a large set of subcommands and an EXEC 2 or REXX subcommand calling environment.

The XMEDIT subcommands provide commands for all of the functions you can do interactively with the menu editor, such as adding data, adding fields, naming fields and so on. In addition, a large set of subcommands have been added to retrieve information about the menu, such as its data, fields, size, etc.

These subcommands can be specified in several ways:

1. They can be used in XMEDIT macros—EXEC 2 or REXX EXECs with a filetype of XMEDIT. By combining these subcommands, you can write sophisticated editing functions that can perform an almost unlimited variety of editing tasks, such as naming fields in a particular order, drawing boxes around an area, moving default data into a new menu, creating stripes of colored attributes for effect, loading consistent-look field positions and attributes, changing all attributes of one type to another, and so on. There are also subcommands allowing you to see how the menu is progressing while in the macro, giving you the ability to design live demonstrations.

   One special macro, usually named PROFILE, is implicitly called before editing begins. This macro can be used to customize your editing environment and set PF keys. You can also design special-purpose profile files to pre-initialize a menu to corporate standards, or convert data from a foreign format to XMENU format, for example. See Appendix F, "Changing the defaults by editing PROFILE XMEDIT" on page 127, for a sample of this macro file.

2. They can be specified in the input area of the pop-up command menu. This is a pop-up menu with a command line that enables you to enter XMEDIT subcommands, CMS, and CP commands without leaving XMEDIT. You can use this command line to change editing settings, such as changing whether OLDWAY is active or not.

3. They can be assigned to Input mode and command menu PF keys. This is how Input mode keys are defined. From XMEDIT Input mode, for example, you can press PF13 to display the PF key definition menu—the result of the `QUERY SCREEN INPUT` command. Simply type over any command associated with the PF or PA keys on this display with your choice of XMEDIT subcommands.

XMEDIT subcommands

Many XMEDIT subcommands simulate user input into XMEDIT Input mode. Many are also based on, or modify the position of, XMEDIT's Edit mode cursor.

XMEDIT subcommands are presented here in alphabetical order.

BACKSPAC subcommand

The BACKSPAC subcommand moves the editing cursor one or more characters backward (toward the top left corner of the menu).

The format of the BACKSPAC subcommand is shown below:

BACKspac	[*nn*]

Where

nn The number of positions to move left. If not specified, *nn* defaults to one.

Usage notes

This subcommand is used to reposition the editing cursor. You can set a PF key to a specific value to use the key as a space counter or a back tab key.

Return codes

1 Invalid decimal number passed.
2 Value passed exceeds the menu size.
3 Value passed is less than zero.

BLANK subcommand

The BLANK subcommand sets one or more characters in the menu to blanks, starting at the current position of the editing cursor.

The format of the BLANK subcommand is shown below:

BLank	[*nn*]

Where

nn The number of positions to convert to blanks. If not specified, *nn* defaults to one.

Usage notes

This subcommand is used to set an area of the menu to blanks. Any fields within the blanked area are deleted.

If the count causes the blanking process to exceed the bottom of the menu, the blanking process continues at the top left corner of the menu. Therefore, BLANK 1920 would completely clear a 24 row by 80 column menu, no matter where the cursor is.

The editing cursor is repositioned to the space immediately following the blanked area.

Return codes

1 Invalid decimal number passed.

CANCEL subcommand

The CANCEL subcommand immediately exits the XMEDIT editing session, whether or not any changes have been made, without first prompting for acknowledgement.

The format of the CANCEL subcommand is shown below:

```
CANCel
```

Usage notes

This subcommand is used to immediately end an XMEDIT session without saving the menu or showing the QUIT confirmation pop-up menu.

This subcommand can be used to end "canned" sessions, such as automatic menu generation processes or online full-screen demonstrations.

CANCEL is synonymous with QQUIT.

COMMAND subcommand

The COMMAND subcommand displays a pop-up menu allowing you to enter XMEDIT subcommands, CMS, or CP commands.

The format of the COMMAND subcommand is shown below:

```
Command
```

Usage notes

This subcommand is used to display a pop-up menu allowing you to enter XMEDIT subcommands, CMS, or CP commands.

Enter each subcommand or command on the pop-up menu, then press ENTER. A question mark ("?") redisplays previously entered commands; an equals sign ("=") repeats the last entered command. Prefixing a command with an ampersand ("&") leaves the command in the input area, allowing for its repeated use.

When each command completes, its non-zero return code is displayed to the left of the input area. If a non-zero return code is returned, the command remains in the input area, and the cursor is placed immediately following it to allow for corrections. Move the cursor to the beginning of the command area and press ERASE EOF to remove the command.

The QUIT command or PF3/PF15 will remove the COMMAND pop-up menu.

In Input mode, COMMAND is assigned by default to PF1.

CURSOR subcommand

The CURSOR subcommand sets the initial position of the cursor when the menu is displayed to the current position of the editing cursor.

The format of the CURSOR subcommand is shown below:

```
CURsor
```

Usage notes

This subcommand sets the cursor position when the menu is displayed in an application (assuming the application hasn't explicitly moved it).

The current editing cursor position is used as the display cursor position. To specifically position the display cursor, use the POSITION subcommand followed by the CURSOR subcommand. For example,

```
POSITION 20 20
CURSOR
```

would position the cursor on row 20, column 20.

CURSOR is synonymous with IC.

DATA subcommand

The DATA subcommand moves data to the menu starting at the editing cursor position. Leading blanks are truncated.

The format of the DATA subcommand is shown below:

DATA	*string*

Where

string The character string to be moved to the menu.

Usage notes

This subcommand moves data to the menu. Any overlaid fields are destroyed.

The editing cursor is moved to the space immediately following the data moved to the menu. Therefore, sequential DATA subcommands move contiguous data to the menu.

If excess data exists when the bottom of the menu is reached, data movement continues at the top left corner of the menu.

Leading blanks are truncated. Use BLANK or DDATA if you need to move leading blanks to the menu.

Use the current NULLCHAR character to move NULLS to the menu. You can determine the setting of the NULLCHAR by issuing the subcommand QUERY NULLCHAR.

Return codes

14 No data passed.

DDATA subcommand

The DDATA subcommand moves data to the menu starting at the editing cursor position. Leading blanks are moved.

The format of the DDATA subcommand is shown below:

DDATA	*delim string*

Where

delim Any delimiting character.

string The character string to be moved to the menu.

Usage notes

This subcommand moves data to the menu. Any overlaid fields are destroyed.

The editing cursor is moved to the space immediately following the data moved to the menu. Therefore, sequential DDATA subcommands move contiguous data to the menu.

If excess data exists when the bottom of the menu is reached, data movement continues at the top left corner of the menu.

Leading blanks (blanks following the delimiter character) are moved to the menu.

No trailing delimiter is needed or recognized.

Use the current NULLCHAR character to move NULLS to the menu. You can determine the setting of the NULLCHAR by issuing the subcommand QUERY NULLCHAR.

Return codes

14 No data passed.

DESTROY subcommand

The DESTROY subcommand destroys (removes) a field from the menu, or, if a field does not exist at the current cursor position, moves the editing cursor back to the previous field marker.

The format of the DESTROY subcommand is shown below:

```
DESTRoy
```

Usage notes

This subcommand is used to remove a field from the menu. The current position of the editing cursor determines the field to be removed.

If the editing cursor is not positioned on a field, the cursor is moved backward until it reaches a previous field marker. If none are found before the top of the menu, searching continues at the bottom right corner of the menu. If none are found, the cursor position remains unchanged.

By default, PA2 is set to DESTROY in Input mode.

DISPLAY subcommand

The DISPLAY subcommand displays the menu as it will look when displayed by an application.

The format of the DISPLAY subcommand is shown below:

DISPlay	[NOWAIT]

Where

NOWAIT Specifies that the menu is displayed and return is made directly to the subcommand caller. If NOWAIT is not specified, the menu is displayed until you press an interrupt-producing key.

Usage notes

This subcommand displays the menu as it will appear in your application program. This subcommand allows you to see how your menu will look to an end user.

The DISPLAY subcommand places you into Display mode. If you press PA2 when the menu is displayed, the menu and a list of its fields, their names and attributes are printed, then the subcommand returns to its caller. Pressing any other key also returns to the subcommand caller.

The NOWAIT parameter allows you to display the menu and return to the subcommand caller without waiting for user input. This enables you to generate XMEDIT macros used for menu demonstration purposes.

In Input mode, DISPLAY (without parameters) is assigned to PA1 by default.

DOWN subcommand

The DOWN subcommand scrolls the Input and Display mode windows one or more lines downward, that is, lines below the displayed area are moved into the displayed area.

The format of the DOWN subcommand is shown below:

Down	[*nn*]

Where

nn The number of lines to move the Input and Display mode window down. If not specified, the counter defaults to one.

Usage notes

This subcommand is used to reposition the Input and Display modes when the menu you are editing is larger than the terminal screen size you are using to display it.

Return codes

1 Invalid decimal number passed.

EDIT subcommand

The EDIT subcommand places you in XMEDIT Edit mode.

The format of the EDIT subcommand is shown below:

```
Edit
```

Usage notes

This subcommand is used to place you in XMEDIT Edit mode.

Edit mode presents a prefix area which is used to move, add, center, copy, delete, shift, and import menu lines. Enter prefix area commands, then press the ENTER key. PF7/PF19 is used to shift data on the Edit mode screen downward when editing menus larger than your terminal screen size. PF8/PF20 is used to shift data on the Edit mode screen upward when editing menus larger than your terminal screen size. When in Edit mode, press PF1/PF13 to get Edit mode-specific help. Press PF3/PF15 to return to the subcommand caller.

In Input mode, the ENTER key is normally set to EDIT

FIELD subcommand

The FIELD subcommand creates or updates a field on the menu.

The format of the FIELD subcommand is shown below:

FIeld	[PROTect|UNPROTect|NUMERIC] [BRIGHT|DIM|DARK|LGHTPEN] [MDT|NOMDT] [SKIP|NOSKIP] [BLUE|GREEN|RED|YELLOW|PINK|TURQuoise| WHITE|DEFCOL] [BLINK|UNDERscore|REVerse|NOHIlight] [PS *nn*] [NAMe *fieldname*]

Where

PROTect|UNPROTect|NUMERIC
Specify the field's protection attributes.

BRIGHT|DIM|DARK|LGHTPEN
Specify the field's intensity.

MDT|NOMDT Specify whether the field's Modified Data Tag (MDT) is set.

SKIP|NOSKIP Specify whether the field's skip bit is set (protected fields only).

BLUE|GREEN... Specify the field's extended color.

BLINK|UNDERscore...
Specify the field's extended highlighting.

PS *nn* Specifies the field's symbol set. *Nn* must be a hexadecimal value of X'00', or X'40' through X'EF'.

NAMe *fieldname* Specifies that the field have a logical name of *fieldname*.

Usage notes

This subcommand is used to create a field at the position of the editing cursor. If no parameters are specified, a pop-up menu is displayed, allowing you to interactively specify the field's attributes.

Multiple attribute values can be specified in the subcommand. If multiple attribute definitions for one attribute type exist, the last one specified is used.

The MDT parameter is used to return a field as input even if the user does not modify it. This allows you to create menus with default data returned as user input even if the user did not enter it.

The NUMERIC parameter implies both unprotected and numeric.

The LGHTPEN parameter implies both DIM and selector pen detectable. BRIGHT is always selector pen detectable, and DARK is never selector pen detectable.

If a programmed symbol set is specified, it must be a hexadecimal value of zero, or X'40' through X'EF' (X'F1' cannot be specified for a field attribute).

Return codes

7 Field name is too long.
8 Field name is invalid or missing.
10 Field name already exists.
11 Cannot add any more fields to this menu.
12 Invalid attribute.
13 Invalid or missing symbol set number.

FILE subcommand

The FILE subcommand saves the current state of the menu to disk and immediately exits the XMEDIT editing session.

The format of the FILE subcommand is shown below:

FILE	[*fname* [*ftype* [*fmode*]]]

Where

fname The CMS filename of the file to be saved. If not specified, *fname* defaults to the name of the menu being edited.

ftype The CMS filetype of the file to be saved. If not specified, *ftype* defaults to MENU.

fmode The CMS filemode of the file to be saved. If not specified, *fmode* defaults to A2.

Usage notes

This subcommand is used to save your menu editing work and end your XMEDIT session.

If AUTONAME was specified when you invoked XMEDIT, then fields will be automatically named before the menu is saved to disk.

If the NAME option was specified when you invoked XMEDIT, you will be prompted for each field's name before the menu is saved to disk.

If the DSECT, MENTEXT, or TEXT options were specified, the appropriate COPY and/or object files will be generated.

Return codes

If an error occurs attempting to save the menu to disk, an error message, including the CMS file system return code, will be displayed and the save will be aborted.

FILL subcommand

The FILL subcommand sets one or more characters in the menu to a fill character, starting at the current position of the editing cursor.

The format of the FILL subcommand is shown below:

FILL	*char [nn]*

Where

char The fill character.

nn The number of positions to fill with characters. If not specified, the counter defaults to one.

Usage notes

This subcommand is used to fill an area of the menu with a specific character. Any fields within the filled area are deleted.

If the counter causes the fill process to extend beyond the bottom right corner of the menu, the fill process continues at the top left corner of the menu.

POSITION 1 1 followed by FILL * 80 would fill the entire top line of a 80 column menu to asterisks.

The editing cursor is positioned to the space immediately following the filled area.

Use the current NULLCHAR character to move NULLS to the menu. You can determine the setting of the NULLCHAR by issuing the subcommand QUERY NULLCHAR.

Return codes

1 Invalid decimal number passed.

FWDSPACE subcommand

The FWDSPACE subcommand moves the editing cursor one or more characters forward toward the bottom right corner of the menu.

The format of the FWDSPACE subcommand is shown below:

FWDspace	[*nn*]

Where

nn The number of positions to move right. If not specified, the counter defaults to one.

Usage notes

This subcommand is used to reposition the editing cursor. You can set a PF key to a specific value and then use the key as a space counter or a tab key.

Return codes

1 Invalid decimal number passed.
2 Value passed exceeds the menu size.
3 Value passed is less than zero.

HELP subcommand

The HELP subcommand calls the CMS HELP command.

The format of the HELP subcommand is shown below:

Help	[*help-parms*]

Where

help-parms Parameters normally passed to the CMS HELP command. If not specified, the CMS command HELP XMEDIT MENU is invoked.

Usage notes

This subcommand is used to display XMEDIT online help screens.

HELP is normally set to the PF13 key when in Input mode.

HELPMENU subcommand

The HELPMENU subcommand displays the default Input mode PF key definitions.

The format of the HELPMENU subcommand is shown below:

```
HELPMenu
```

Usage notes

This subcommand is used to display a menu showing the default XMEDIT Input mode PF Key definitions.

HELPMENU can be set to a PF key when in Input mode to give fast access to the default key settings, although we recommend instead setting a key to the subcommand QUERY SCREEN INPUT, which is both more accurate (if you've redefined keys), and allows you to change key definitions; QUERY SCREEN INPUT is set by default in Input mode to PF13.

IC subcommand

The IC subcommand sets the initial position of the cursor when the menu is displayed to the current position of the editing cursor.

The format of the IC subcommand is shown below:

```
IC
```

Usage notes

This subcommand sets the cursor position when the menu is displayed in an application (assuming the application hasn't explicitly moved it).

The current editing cursor position is used as the display cursor position. To specifically position the display cursor, use the POSITION subcommand followed by the IC subcommand. For example,

```
POSITION 20 20
IC
```

would position the cursor on row 20, column 20.

IC is synonymous with CURSOR.

LEFT subcommand

The LEFT subcommand scrolls the Input and Display mode windows one or more positions left, that is, the area to the right of the displayed area is moved into the displayed area.

The format of the LEFT subcommand is shown below:

Left	[*nn*]

Where

nn The number of positions to move the Input and Display mode window left. If not specified, the counter defaults to one.

Usage notes

This subcommand is used to reposition the Input and Display modes when the menu you are editing is larger than the terminal screen size you are using to display it.

Return codes

1 Invalid decimal number passed.

LOWCASE subcommand

The LOWCASE subcommand converts all characters from the editing cursor position to the next blank or null character to lowercase.

The format of the LOWCASE subcommand is shown below:

```
LOwcase
```

Usage notes

This subcommand is used to convert a string of characters to lowercase. Conversion continues until a blank or null is detected.

The editing cursor is moved to the space immediately following the converted string.

NAME subcommand

The NAME subcommand is used to name menu fields, or, if a field does not exist at the current cursor position, to move the editing cursor forward to the next field marker.

The format of the NAME subcommand is shown below:

Name	[*name* | NONAME]

Where

name The name of the field. The name must be one to seven characters; its first letter must be alphabetic. Letters, numbers, and the characters "#", "@", "!", "?", "$", "¢", underscore, and period are valid name characters.

NONAME Specifies that the field be unnamed. If it is already named, the name is removed.

Usage notes

This subcommand names or removes the name from a menu field. If no parameter is passed, a pop-up menu is displayed, allowing you to enter a field name.

To use the pop-up menu, enter a valid field name, then press ENTER. To remove the field name, press ERASE EOF, then ENTER.

If PF12/PF24 is pressed while in the Name-a-field pop-up menu, the cursor is moved to the next field marker. If none are found before the bottom of the menu, searching continues at the top of the menu. If none are found, the cursor remains where it started.

In Input mode, NAME (without parameters) is assigned to PF12/PF24 by default.

Return codes

7 Field name exceeds seven characters.
8 Field name is invalid.
9 Editing cursor not on a field marker.
10 Field name already exists.

NULL subcommand

The NULL subcommand sets one or more characters in the menu to nulls, starting at the current position of the editing cursor.

The format of the NULL subcommand is shown below:

NULl	[*nn*]

Where

nn The number of positions to convert to nulls. If not specified, the counter defaults to one.

Usage notes

This subcommand is used to set an area of the menu to nulls. Any fields within the nulled area are deleted.

If the counter causes the nulling process to extend past the bottom right corner of the menu, the nulling process continues at the top left corner of the menu. Therefore, NULL 1920 would completely clear a 24 row by 80 column menu, no matter where the cursor is.

The editing cursor is repositioned to the space immediately following the nulled area.

A NULLCHAR character must be in effect to use this subcommand.

Return codes

1 Invalid decimal number passed.
5 No NULLCHAR set.

POSITION subcommand

The POSITION subcommand sets the position of the editing cursor.

The format of the POSITION subcommand is shown below:

POSition	*offset* *row column*

Where

offset The position, relative to the upper left corner of the menu, to place the editing cursor. The top left corner is position zero.

row The row to position the editing cursor. The top row is row one.

column The column to position the editing cursor. The left column is column one.

Usage notes

This subcommand is used to position the editing cursor. Use this subcommand to explicitly position the editing cursor before using other subcommands such as DATA, DDATA, FIELD, or QUERY.

Return codes

1 Invalid decimal number passed.
2 Value passed exceeds the menu size.
16 Row value exceeds the menu size.
17 Column value exceeds the menu size.

PRINT subcommand

The PRINT subcommand prints a copy of the menu to the virtual printer or to a CMS file.

The format of the PRINT subcommand is shown below:

PRint	[MENU [MAP]] [INPUT]

Where

MENU Specifies that the menu is printed.

MAP Specifies that a field map containing field positions, names, and attribute values is printed.

INPUT Specifies that a copy of the Input mode window is printed.

Usage notes

This subcommand is used to print the edited menu and menu information. If no parameters are specified, the menu and the field map are printed.

PRINT is synonymous with PSCREEN.

Return codes

15 Invalid parameter(s) passed.

PSCREEN subcommand

The PSCREEN subcommand prints a copy of the menu to the virtual printer or to a CMS file.

The format of the PSCREEN subcommand is shown below:

PSCreen	[MENU [MAP]] [INPUT]

Where

MENU Specifies that the menu is printed.

MAP Specifies that a field map containing field positions, names, and attribute values is printed.

INPUT Specifies that a copy of the Input mode window is printed.

Usage notes

This subcommand is used to print the edited menu and menu information. If no parameters are specified, the menu and the field map are printed.

PSCREEN is synonymous with PRINT.

Return codes

15 Invalid parameter(s) passed.

QQUIT subcommand

The QQUIT subcommand immediately exits the XMEDIT editing session, whether or not any changes have been made, without first prompting for acknowledgement.

The format of the QQUIT subcommand is shown below:

```
QQuit
```

Usage notes

This subcommand is used to immediately end an XMEDIT session without saving the menu or showing the QUIT confirmation pop-up menu.

This subcommand can be used to end "canned" sessions, such as automatic menu generation processes or online full-screen demonstrations.

QQUIT is synonymous with CANCEL.

QUERY subcommand

The QUERY subcommand places information into the CMS system stack. All stacked lines begin with an asterisk followed by a blank, so that unread stacked lines are treated as comments and ignored.

The format of the QUERY subcommand is shown below:

Query	ALT CHAR CURsor DATA [*count*] ENTER EXT FIELD KEYp LINENd MDT MODE NAME OLDWAY POSition SCReen INPUT|COMMAND [*key*] SIZE SKIP TESTREQ UPCase

Where

ALT Stacks the setting of Alternate Input mode, either OFF or ON. For example:

```
* OFF
```

CHAR Stacks the unnamed field marker character, the named field marker character, and the NULL character. For example:

```
* ¦ ~ `
```

CURsor Stacks the row position, the column position, and the offset of the display cursor position. For example:

```
* 13 33 992
```

DATA *count*

Stacks *count* characters of menu data starting at the position of the editing cursor. If *count* isn't specified, it defaults to one. For example, QUERY DATA 9 might return the following:

```
* Some Data
```

ENTER Stacks the setting of the ENTER key in XMEDIT Input mode. For example:

```
* SCREEN INPUT ENTE EDIT
```

EXT Stacks whether the menu has extended attributes. OFF means the menu has no extended attributes, ON means it does. For example:

```
* OFF
```

FIELD Stacks the name and attributes of the field marker on which the editing cursor is positioned. If the cursor isn't on a field marker, a return code of 9 is returned. Data is returned in the following order: field name (or NONAME if the field isn't named), protection, intensity, SKIP, MDT, extended color, extended highlighting, and programmed symbol set. If the pseudo field starting at the top of the menu is queried, only the field name is returned. An example of QUERY FIELD responses follows:

```
* FNAME PROT BRIGHT NOSKIP NOMDT YELLOW BLINK 00
```

KEYp Stacks the last interrupt-generating key pressed in Input mode. For example:

```
* PF12
```

LINENd Stacks whether a command separation character exists, and what it is. For example:

```
* ON #
```

MDT Stacks the setting of MDT, either ON or OFF. ON means it was specified, OFF means it wasn't. For example:

```
* OFF
```

MODE Stacks whether this macro was called in Input mode or Edit mode. Either INPUT or EDIT is returned. For example:

```
* INPUT
```

NAME Stacks the name of the menu. For example:

```
* PVM
```

OLDWAY Stacks the setting of OLDWAY, either ON or OFF. ON means it was specified, OFF means it wasn't. For example:

```
* OFF
```

POSition Stacks the row position, the column position, and the offset of the editing cursor position. For example:

```
* 6 32 431
```

SCREEN INPUT|COMMAND *key*

If the optional *key* parameter is not specified, this command presents a full-screen display of the current key definitions for either Input mode or the COMMAND pop-up menu, where you can type over an existing key definition to change its function. Otherwise, this command stacks the setting of the specified PF key for either Input mode or the COMMAND pop-up menu. For example QUERY SCREEN INPUT PF04 could return the following information to the stack:

```
* SCREEN INPUT PF04 FIELD PROT BRIGHT
```

SIZE Stacks the size of the menu being edited. The number of rows, columns, and characters are stacked. For example:

```
* 24 80 1920
```

SKIP Stacks the setting of SKIP, either ON or OFF. ON means it was specified, OFF means it wasn't. For example:

```
* OFF
```

TESTREQ Stacks the setting of the TESTREQ key in XMEDIT Input mode. For example:

```
* SCREEN INPUT TEST DISPLAY
```

UPCase Stacks the setting of UPCASE, either ON or OFF. ON means it was specified, OFF means it wasn't. For example:

```
* OFF
```

Usage notes

This subcommand is used to retrieve information about the XMEDIT session. It is typically used within an XMEDIT macro to test for various conditions; for example, you could develop an XMEDIT macro to simulate a user demonstration and perform various functions based on which PF key is pressed. This information can also be used to save the existing environment before making changes; for example, you can preserve the current editing cursor position, perform some work, then restore it with the SET subcommand.

Unrecognized QUERY subcommands are passed to CMS and CP for processing.

QUERY is synonymous with TRANSFER.

Return codes

1 Invalid decimal number passed.
9 Cursor is not on a field marker.

QUIT subcommand

The QUIT subcommand exits the XMEDIT editing session, whether or not any changes have been made, but first prompts for acknowledgement.

The format of the QUIT subcommand is shown below:

```
QUIT
```

Usage notes

This subcommand is used to end an XMEDIT session without saving the menu.

A pop-up menu is displayed, allowing you to either continue the editing session or indeed terminate it. When this menu is displayed, pressing PF2/PF14 causes XMEDIT to quit, any other key allows XMEDIT to continue.

If QUIT is entered from the COMMAND pop-up menu, only the pop-up menu exits: the XMEDIT editing session continues.

By default, in Input mode, QUIT is assigned to PF2/PF14.

REFRESH subcommand

The REFRESH subcommand momentarily displays the Input mode display.

The format of the REFRESH subcommand is shown below:

```
REFresh
```

Usage notes

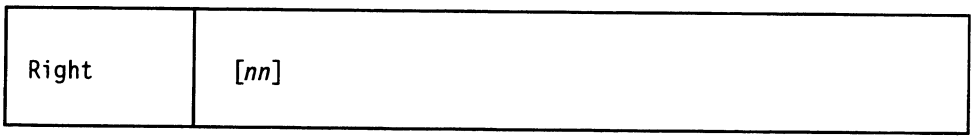
This subcommand is normally used within an XMEDIT macro to display the progress of that XMEDIT macro. The current state of Input mode is displayed, then the macro continues.

This subcommand can be used to generate XMEDIT usage demonstrations.

RIGHT subcommand

The RIGHT subcommand scrolls the Input and Display mode windows one or more positons right, that is, the area to the left of the displayed area is moved into the displayed area.

The format of the RIGHT subcommand is shown below:



Where

nn The number of positions to move the Input and Display mode window right. If not specified, the counter defaults to one.

Usage notes

This subcommand is used to reposition the Input and Display modes when the menu you are editing is larger than the terminal screen size you are using to display it.

Return codes

1 Invalid decimal number passed.

SAVE subcommand

The SAVE subcommand saves the current state of the menu to disk.

The format of the SAVE subcommand is shown below:

SAVE	[*fname* [*ftype* [*fmode*]]]

Where

fname The CMS filename of the file to be saved. If not specified, *fname* defaults to the name of the menu being edited.

ftype The CMS filetype of the file to be saved. If not specified, *ftype* defaults to MENU.

fmode The CMS filemode of the file to be saved. If not specified, *fmode* defaults to A2.

Usage notes

This subcommand is used to save your menu editing work without ending your XMEDIT session.

If AUTONAME was specified when you invoked XMEDIT, then fields will be automatically named before the menu is saved to disk.

If the NAME option was specified, you will be prompted for each field's name before the menu is saved to disk.

If either the DSECT, MENTEXT, or TEXT options were specified, the appropriate COPY and/or object decks will be generated.

Return codes

If an error occurs attempting to save the menu to disk, an error message, including the CMS file system return code, will be displayed and the save will be aborted.

SET subcommand

The SET subcommand is used to modify the XMEDIT editing environment.

The format of the SET subcommand is shown below:

SET	ALT OFF|ON CHAR *char1 char2* ENTER *string* EXT OFF|ON LINENd OFF|ON *char* MDT OFF|ON NULLchar *char3* OLDWAY OFF|ON SCREEN INPUT|COMMAND *key string* SKIP OFF|ON TESTREQ *string* UPCase OFF|ON

Where

ALT Specifies whether Alternate Input mode is being used. If ON, it is to be used; if OFF, it isn't. The default is OFF. Here is an example of this subcommand's use:

```
SET ALT ON
```

CHAR *char1 char2*

Specifies the characters used to mark named and unnamed fields. *Char1* is the character used to mark named fields. *Char2* is the character used to mark unnamed fields. *Char1* and *char2* must be different. Each can be specified as either a character or as a two-digit hexadecimal number. The defaults are "¬" (X'A1') and "¦" (X'6A'). Here is an example of the subcommand's use:

```
SET CHAR > +
```

ENTER *string*

Specifies the command executed when the ENTER key is pressed. The default Input mode ENTER key is set to enter Edit mode. Here is an example of this subcommand's use:

```
SET ENTER DISPLAY
```

EXT Specifies whether extended attributes are allowed in the menu. If ON, extended attributes will be allowed; if OFF, they won't be. This command can only be used in the XMEDIT profile, and is, by default, set to the attribute capabilities of the terminal being used to create the menu. Here is an example of this subcommand's use in PROFILE XMEDIT:

```
⋮
SET EXT ON
⋮
```

LINENd OFF|ON *char*

Specifies whether an end-of-command character exists and what it is. If ON, the end-of-command character separates commands typed on the same line. If OFF, no end-of-command character exists. By default, the character is # and is set ON. Here is an example of the subcommand's use:

```
SET LINEN OFF
```

MDT Specifies whether newly-created unprotected fields have their Modified Data Tag (MDT) set. If ON, MDT bits are set; if OFF, MDT bits aren't set. The default setting is OFF. Here is an example of this subcommand's use:

```
SET MDT ON
```

NULLchar *char3*

Specifies the characters used to mark NULL (binary zero) characters in the menu. It can be specified as either a character or as a two-digit hexadecimal number. The default NULL character is "`" (X'79'). Here is an example of this subcommand's use:

```
SET NULL \
```

OLDWAY Specifies whether pop-up menus are used for defining attributes and naming fields. If ON, the "old-style" XMENU menus are used; if OFF, XMEDIT pop-up menus are used. When using XMEDIT, the default setting is OFF. Here is an example of this subcommand's use:

```
SET OLDWAY ON
```

SCREEN INPUT|COMMAND *key string*

Specifies the command executed when the given key is pressed under the given screen. INPUT signifies keys pressed while in XMEDIT Input mode. COMMAND signifies keys pressed while in the COMMAND pop-up menu. Here is an example of the subcommand's use:

```
SET SCREEN INPUT PF09 FIELD PROT SKIP BRIGHT RED REVERSE
```

SKIP Specifies whether newly-created protected fields automatically have their SKIP bit set. If ON, SKIP bits are set; if OFF, SKIP bits aren't set. The default setting is OFF. Here is an example of the subcommand's use:

```
SET SKIP ON
```

TESTREQ *string*

Specifies the command executed when the TEST REQUEST key is pressed. This key can only be used on local, channel-attached terminals. Here is an example of the subcommand's use:

```
SET TESTREQ DISPLAY
```

UPCase Specifies whether the entire menu is converted to uppercase before being saved to disk. If ON, the menu is converted; if OFF, it isn't. The default setting is OFF. Here is an example of the subcommand's use:

```
SET UPC ON
```

Usage notes

These subcommands are used to modify the XMEDIT editing environment. They can be used from XMEDIT macros or from the COMMAND pop-up menu command line.

You can use SET OLDWAY to switch between "old style" menus and new pop-up menus without leaving the XMEDIT session.

You can use SET SCREEN INPUT to dynamically change the functions of your Input mode PF keys.

You can create a profile macro that establishes a customized environment for your work. This file is called PROFILE XMEDIT by default.

Return codes

15 Invalid parameter(s) passed.
18 Subcommand only valid in the XMEDIT profile.

TABFLDB subcommand

The TABFLDB subcommand moves the editing cursor back to the previous field marker.

The format of the TABFLDB subcommand is shown below:

```
TABFLDB
```

Usage notes

This subcommand is used to move the cursor to the previous field marker. If none are found before the top left corner of the menu, searching continues at the bottom right corner of the menu. If none are found, the cursor remains where it started.

TABFLDB can be set to a PF key when using Alternate Input mode to enable you to tab from field to field.

TABFLDF subcommand

The TABFLDF subcommand moves the editing cursor forward to the next field marker.

The format of the TABFLDF subcommand is shown below:

```
TABfldf
```

Usage notes

This subcommand is used to move the cursor to the next field marker. If none are found before the bottom right corner of the menu, searching continues at the top left corner of the menu. If none are found, the cursor remains where it started.

TABFLDF can be set to a PF key when using Input mode to enable you to tab from field to field.

TRANSFER subcommand

The TRANSFER subcommand places information into the CMS system stack. All stacked lines begin with an asterisk followed by a blank, so that unread stacked lines are treated as comments and ignored.

The format of the TRANSFER subcommand is shown below:

TRANSFER	ALT CHAR CURsor DATA [*count*] ENTER EXT FIELD KEYp LINENd MDT MODE NAME OLDWAY POSition SCReen INPUT|COMMAND [*key*] SIZE SKIP TESTREQ UPCase

Where

ALT Stacks the setting of Alternate Input mode, either OFF or ON. For example:

```
* OFF
```

CHAR Stacks the unnamed field marker character, the named field marker character, and the NULL character. For example:

```
* ¦ - `
```

CURsor Stacks the row position, the column position, and the offset of the display cursor position. For example:

```
* 13 33 992
```

DATA *count*

Stacks *count* characters of menu data starting at the position of the editing cursor. If *count* isn't specified, it defaults to one. For example, TRANSFER DATA 9 might return the following:

```
* Some Data
```

ENTER Stacks the setting of the ENTER key in XMEDIT Input mode. For example:

```
* SCREEN INPUT ENTE EDIT
```

EXT Stacks whether the menu has extended attributes. OFF means the menu has no extended attributes, ON means it does. For example:

```
* OFF
```

FIELD Stacks the name and attributes of the field marker on which the editing cursor is positioned. If the cursor isn't on a field marker, a return code of 9 is returned. Data is returned in the following order: field name (or NONAME if the field isn't named), protection, intensity, SKIP, MDT, extended color, extended highlighting, and programmed symbol set. If the pseudo field starting at the top of the menu is queried, only the field name is returned. An example of TRANSFER FIELD responses follows:

```
* FNAME PROT BRIGHT NOSKIP NOMDT YELLOW BLINK 00
```

KEYp Stacks the last interrupt-generating key pressed in Input mode. For example:

```
* PF12
```

LINENd Stacks whether a command separation character exists, and what it is. For example:

```
* ON #
```

MDT Stacks the setting of MDT, either ON or OFF. ON means it was specified, OFF means it wasn't. For example:

```
* OFF
```

MODE Stacks whether this macro was called in Input mode or Edit mode. Either INPUT or EDIT is returned. For example:

```
* INPUT
```

NAME Stacks the name of the menu. For example:

```
* PVM
```

OLDWAY Stacks the setting of OLDWAY, either ON or OFF. ON means it was specified, OFF means it wasn't. For example:

```
* OFF
```

POSition Stacks the row position, the column position, and the offset of the editing cursor position. For example:

```
* 6 32 431
```

SCREEN INPUT|COMMAND *key*

If the optional *key* parameter is not specified, this command presents a full-screen display of the current key definitions for either Input mode or the COMMAND pop-up menu, where you can type over an existing key definition to change its function. Otherwise, this command stacks the setting of the specified PF key for either Input mode or the COMMAND pop-up menu. For example TRANSFER SCREEN INPUT PF04 could return the following information to the stack:

```
* SCREEN INPUT PF04 FIELD PROT BRIGHT
```

SIZE Stacks the size of the menu being edited. The number of rows, columns, and characters are stacked. For example:

```
* 24 80 1920
```

SKIP Stacks the setting of SKIP, either ON or OFF. ON means it was specified, OFF means it wasn't. For example:

```
* OFF
```

TESTREQ Stacks the setting of the TESTREQ key in XMEDIT Input mode. For example:

```
* SCREEN INPUT TEST DISPLAY
```

UPCase Stacks the setting of UPCASE, either ON or OFF. ON means it was specified, OFF means it wasn't. For example:

```
* OFF
```

Usage notes

This subcommand is used to retrieve information about the XMEDIT session. It is typically used within an XMEDIT macro to test for various conditions; for example, you could develop an XMEDIT macro to simulate a user demonstration and perform various functions based on which PF key is pressed. This information can also be used to save the existing environment before making changes; for example, you can preserve the current editing cursor position, perform some work, then restore it with the SET subcommand.

Unrecognized TRANSFER subcommands are passed to CMS and CP for processing.

TRANSFER is synonymous with QUERY.

Return codes

1 Invalid decimal number passed.
9 Cursor is not on a field marker.

UP subcommand

The UP subcommand scrolls the Input and Display mode windows one or more lines upward, that is, lines above the displayed area are moved into the displayed area.

The format of the UP subcommand is shown below:

Up	[*nn*]

Where

nn The number of lines to move the Input and Display mode window up. If not specified, the counter defaults to one.

Usage notes

This subcommand is used to reposition the Input and Display modes when the menu you are editing is larger than the terminal you are using to display it.

Return codes

1 Invalid decimal number passed.

UPCASE subcommand

The UPCASE subcommand converts all characters from the editing cursor position to the next blank or null character to uppercase.

The format of the UPCASE subcommand is shown below:

```
UPCase
```

Usage notes

This subcommand is used to convert a string of characters to uppercase. Conversion continues until a blank or null is detected.

The editing cursor is moved to the space immediately following the converted string.

Chapter 4. XMENULIB

XMENULIB is a utility used to move menus to and from XMENULIB menu libraries. It is similar in function to the CMS MACLIB and TXTLIB commands.

XMENULIB command format

The format of the XMENULIB command is shown below:

XMENULIB	GEN *libname menuname1 [menuname2 ...]* ADD *libname menuname1 [menuname2 ...]* DEL *libname menuname1 [menuname2 ...]* EXTRACT *libname menuname1 [menuname2 ...]* REP *libname menuname1 [menuname2 ...]* COMP *libname* MAP *libname* [([TERM|DISK|PRINT] [)]] ?

Where

GEN *libname menuname1 [menuname2 ...]*
Generates an XMENULIB library called *libname* consisting of *menuname1*, *menuname2*, etc. If the library already exists, it is erased and a new library is created.

ADD *libname menuname1 [menuname2 ...]*
Adds menus to an existing XMENULIB.

DEL *libname menuname1 [menuname2 ...]*
Deletes menus from an existing XMENULIB.

EXTRACT *libname menuname1 [menuname2 ...]*
Moves menus from an existing XMENULIB to separate CMS MENU files.

REP *libname menuname1 [menuname2 ...]*
Replaces existing menus in an existing XMENULIB with menus from existing CMS MENU files.

COMP *libname*
Pretends to compress an existing XMENULIB. Actually, XMENULIB files are always compressed; this option is provided for transparent compatibility with the CMS library command MACLIB.

MAP *libname* [(**TERM|DISK|PRINT]**
Creates a listing of the members of an existing XMENULIB. If either DISK or no option is specified, the output is written to a CMS file named *libname* MAP. If TERM is specified, the listing is displayed on your terminal screen. If PRINT is

specified, the listing is printed. Only one of the three options may be specified.

? Types a short description of the available XMENULIB command options to your screen.

Usage notes

XMENULIB files are collections of XMENU menus. They are similar to MACLIB and TXTLIB files. However, XMENULIBs are variable format files (RECFM V) and may not be compatible with all CMS OS-partitioned dataset simulation.

The XMEDIT editor, the XMENU/REXX Interface (MENUEXEC), and/or the MLOAD and MLOADX subroutines can load menus directly from XMENULIB files. You might want to keep large collections of menus with similar function within one XMENULIB to limit the number of separate menu files on a minidisk. By doing this, you incur some additional overhead loading a menu, but you may save significant overhead by decreasing the number of CMS files (and thus storage pages) that CMS must search when looking for a file on that disk (even when not looking for a menu file specifically).

The XMEDIT editor cannot save menus directly into XMENULIBs; if you want to change a menu within a XMENULIB using the XMEDIT editor you can load it directly, but must save it to a separate CMS file, then replace it using the XMENULIB REP command.

XMENULIB time and date stamps each XMENULIB member. These values are displayed when using the XMENULIB MAP command.

Return codes and messages

1 8310E Insufficient storage to perform XMENULIB function.
2 8306E Menu file is not formatted properly.
3 8311E XMENULIB is not formatted properly.
4 8301E MENU already exists in the library.
4 8309E Member not in XMENULIB.
24 8300E No function specified.
24 8302E No library name specified.
24 8303E Invalid function specified.
24 8304E Invalid parameter specified.
24 8305E Invalid option specified.
24 8318E No MENU filename(s) specified.
28 8307E Menu file caannot be found.
28 8308E Library cannot be found.
100 (Return code following HELP messages.)
xxx 8312E Error writing XMENULIB CMSUT1 temp file.
xxx 8313E Error writing MAP file.
xxx 8314E Error writing MENU file.
xxx 8315E Error writing XMENULIB file.
xxx 8316E Error reading XMENULIB file.
xxx 8317E Error reading MENU file.

Chapter 5. PSEDIT

PSEDIT is a program provided to create and modify 327*x* programmed symbol sets. Both single- and triple-plane symbols are supported.

PSEDIT command format

The format of the PSEDIT command is shown below:

PSEDIT	[*symbol-set-name*|?] [(DISK TRIPLE] [)]

Where

symbol-set-name	The name of a new or existing XMENU programmed symbol set. This is a one-record file having the filetype PSLOAD.
?	Types a short description of the available PSEDIT command options to your screen.
DISK	Specifies that any printed output go to a CMS file instead of to the virtual printer.
TRIPLE	Specifies that you are editing a triple-plane (color) symbol set. This parameter need not be specified when editing an existing triple-plane set.

Creating symbol sets interactively with PSEDIT

Once you enter PSEDIT, you can interactively enter or change each character within the symbol set. You can also display the set as it will appear when displayed if your terminal supports programmed symbol sets.

For single-plane sets, type any character where you want a pixel (picture element) to appear. For triple-plane sets, enter the first letter of the color you want this pixel to appear, for example, "r" for red, "y" for yellow.

Press PF12/PF24 to display the symbol set. If your terminal does not support symbol sets, a message will appear, but you may continue editing the set.

Press PF3/PF15 to save your changes to the symbol set; press PF2/PF14 to leave PSEDIT.

It is also possible to copy symbols from one code point to another. Simply fill in the PSEDIT editing screen and press PF5/PF17. See "PF keys used with PSEDIT" on page 110 for more details on PF key functions available from this screen.

The available matrix for entering symbol set characters is nine dots by sixteen dots. The 3278 model 4 and all 3279 models only support a nine-by-twelve dot matrix. Dots placed into the matrix below line 12 are ignored if displayed on a 3279 or 3278 model 4.

Depending on the size of a 3290 screen, characters loaded in a symbol set may appear truncated to the right and/or bottom. This is because the hardware defines a smaller character size based on the number of characters on the screen. The same characters, displayed in "zoom" mode, will appear in their entirety. Before creating a symbol set, be sure to ascertain the minimum character cell size for the terminals under which your application will run.

Characters can be defined from hexadecimal X'41' to X'FE' (190 characters per set).

If you wish to create a new set of characters using an existing symbol set, copy the existing symbol set to a new file with the filename you wish to give the new set, and a filetype of PSLOAD.

Note: Do not XEDIT a programmed symbol set—this may destroy the file.

Programmed symbol sets can be loaded with the MPSLD subroutine, the XPSLOAD utility, or the PSLOAD or LOADPS MENUEXEC subcommands.

PF keys used with PSEDIT

The following keys are defined for use in PSEDIT.

PF Key	Function
PF2/PF14	Quit (prompts for verification).
PF3/PF15	Saves the updated symbol set and returns to PSEDIT.
PF4/PF16	Prints the current screen image.
PF5/PF17	Copies symbols from one location to another.
PF7/PF19	Scrolls forward to the next character in the symbol set.
PF8/PF20	Scrolls backward to the previous character in the symbol set.
PF12/PF24	Displays the entire symbol set. If your terminal does not support symbol sets, this screen will display the terminal's base character set.

Interactive screens used to develop symbol sets

These figures illustrate the symbol set editor screen and the symbol set display screen.

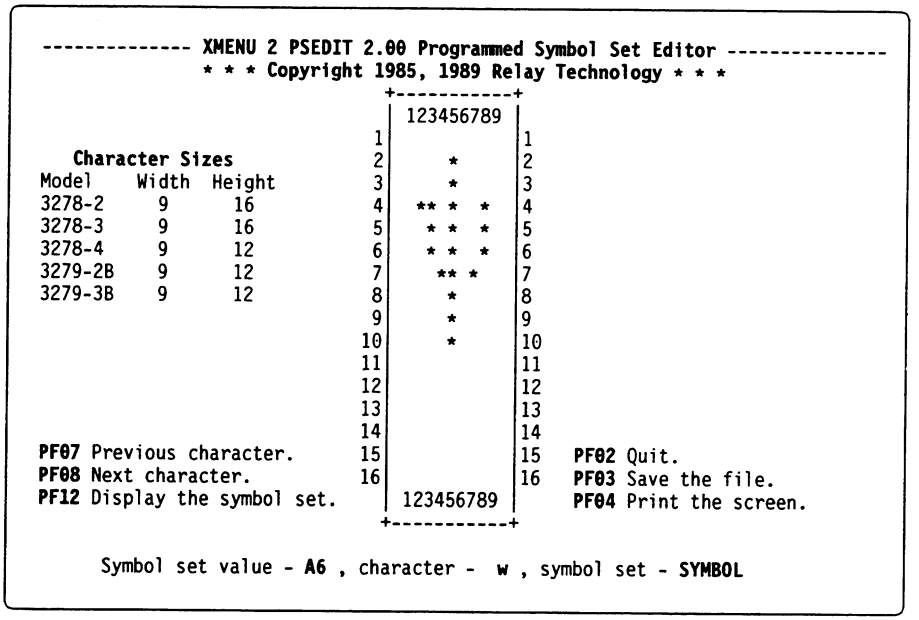


Figure 27. Sample PSEDIT editing screen

To create a symbol, type any non-blank character at each position you want a pixel displayed. When you press ENTER these characters change to asterisks.

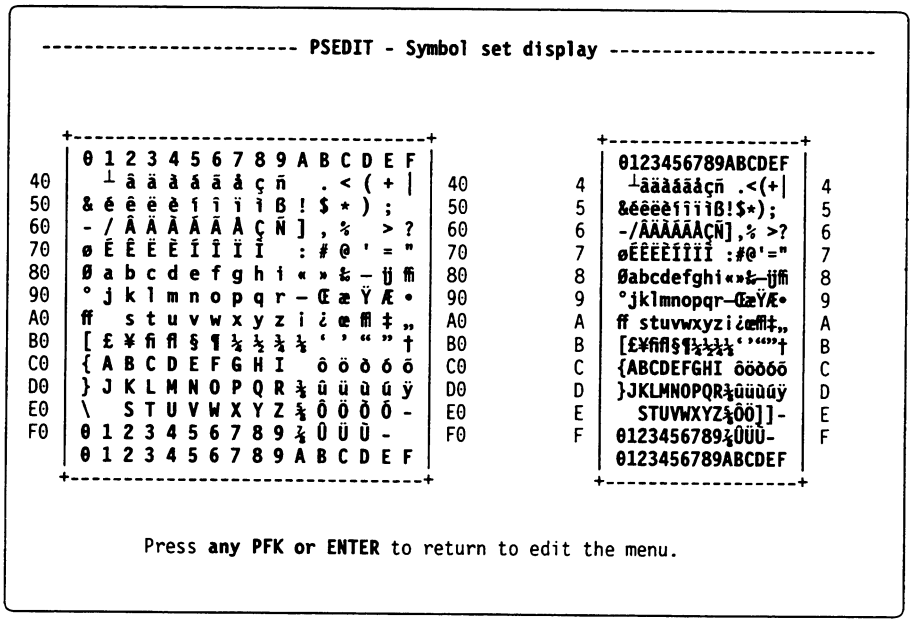


Figure 28. Sample PSEDIT display screen

Return codes and messages

1	8003E PSLOAD file is not properly formatted.
2	8002E Your terminal must be a 3270 to run PSEDIT.
3	8005E Insufficient storage to run PSEDIT.
4	8006E Cannot load PSEDIT menus or your terminal is too small.
100	(Return code following HELP messages.)
xxx	8004E Error reading PSLOAD file.
xxx	8007E Error writing PSLOAD file.
xxx	8008E Error loading symbol set.

Chapter 6. XMENUCOB

The XMENUCOB utility converts the DSECT output of XMEDIT into a COBOL data structure suitable for processing by the MDXSCR subroutine.

XMENUCOB command format

The format of the XMENUCOB command is shown below:

XMENUCOB	*fn [ft [fm [ofn [oft [ofm]]]]]*

Where

fn The name of the file to be converted.

ft The type of the file to be converted. If not specified, it defaults to COPY.

fm The mode of the file to be converted. If not specified, it defaults to *.

ofn The name of the output file. If not specified, it defaults to the name of the input file.

oft The type of the output file. If not specified, it defaults to INCLUDE.

ofm The mode of the output file. If not specified, it defaults to A.

Usage notes

This command allows you to automatically generate a data structure for a COBOL program that uses the MDXSCR subroutine.

This conversion is performed in three steps:

1. Generate the menu using XMEDIT with the DSECT option.
2. Run this utility by issuing the XMENUCOB command.
3. Incorporate the output of this utility into your program. One way to do this is to place this output in a MACLIB and copy it using the COBOL COPY statement.

Return codes and messages

28 Input file is missing.
8 Input file is not the correct format.
xx Error writing output file.

Chapter 7. XMENUPLI

The XMENUPLI utility converts the DSECT output of XMEDIT into a PL/I data structure suitable for processing by the MDXSCR subroutine.

XMENUPLI command format

The format of the XMENUPLI command is shown below:

XMENUPLI	*fn* [*ft* [*fm* [*ofn* [*oft* [*ofm*]]]]]

Where

fn The name of the file to be converted.

ft The type of the file to be converted. If not specified, it defaults to COPY.

fm The mode of the file to be converted. If not specified, it defaults to *.

ofn The name of the output file. If not specified, it defaults to the name of the input file.

oft The type of the output file. If not specified, it defaults to INCLUDE.

ofm The mode of the output file. If not specified, it defaults to A.

Usage notes

This command allows you to automatically generate a data structure for a PL/I program that uses the MDXSCR subroutine.

This conversion is performed in three steps:

1. Generate the menu using XMEDIT with the DSECT option.
2. Run this utility by issuing the XMENUPLI command.
3. Incorporate the output of this utility into your program. One way to do this is to place this output in a MACLIB and copy it using the PL/I %INCLUDE statement.

Return codes and messages

28 Input file is missing.
8 Input file is not the correct format.
xx Error writing output file.

Appendix A. Some notes on menu design

Designing clear and useful menus requires careful thought about the users of your application and what they will be trying to accomplish. It also requires some knowledge of principles of menu design. This section gives a very general overview of some basic guidelines that should be followed when designing menus. Many good books about menu design exist and can be consulted for a more in-depth treatment of menu design considerations. Several books are listed below.

Some guidelines for designing menus

A well-designed menu is:

- Orderly and uncluttered

  Use a lot of white space—blank areas in the menu—so that the menu looks clean and does not overwhelm the user with information.

- Easy to understand

  It should be obvious to users what they must **do** with the menu—what information they must enter, where it must be entered, what keys must be pressed to perform certain actions, etc.

- Sensitive to the user's expectations

  The menus you use in your application give rise to user expectations. If you fail to be consistent within an application, you will violate the user's expectations and create confusion. For example, avoid putting the same piece of data in different locations on different menus.

- Written in plain, everyday English

  Avoid computer jargon and stilted expressions; write as if you were having a conversation with a friend who does not understand what must must be done to accomplish a particular task.

Useful publications

More information on menu design considerations is provided in the following books. You might find one or more of the following helpful:

- **Dumas, Joseph**. *Designing User Interfaces for Software*, Prentice-Hall, Inc. Englewood Cliffs, NJ, 1988, ISBN 0-13-201971-X.
- **Galitz, Wilbert O.** *Handbook of Screen Format Design*, QED Information Sciences, Wellesley, MA, 1985, ISBN 0-89435-119-2.
- **Martin, James**. *Design of Man-Computer Dialogues*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1973, ISBN 0-13-201251-0.
- **Martin, James**. *Application Development Without Programmers*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1982, ISBN 0-13-038943-9.

Appendix B. XMEDIT output files

XMEDIT produces up to five output files. These files are created as a result of the options selected when the XMEDIT editor is invoked and, in two cases—LISTING and SCRIPT—when the PRINT subcommand is also invoked. The output files XMEDIT can create are COPY files, LISTING files, MENU files, SCRIPT files, and TEXT files. The files are described in detail below.

filename **COPY** This file contains an assembler DSECT image of the menu output data. The DSECT name is *filename*. There is an assembler EQU symbol named *filename*S which contains the length of the DSECT in bytes. Fields within the DSECT are labeled with the field names created during menu creation or modification. Lines in the DSECT which contain attributes have comments describing the type of attribute. The header line contains the line and column of the initial cursor position. Each line of the data contains its line and column position. All data is displayed with DC statements, so the file can be used as assembler constants by changing the DSECT statement to a CSECT statement. The *filename* COPY file is only created when either the MENTEXT or COPY XMEDIT option is specified.

filename **MENU** This file contains the menu that is used by the XMENU/REXX Interface program (MENUEXEC) and the XMENU High-Level Language subroutines. It contains variable format records (RECFM V), the first of which contains field control information and field names. The second record contains the actual initial output image of the menu. The specific format of the two records can be found in the XMENU MACLIB member MENUFMT. The virtual storage size of the first record of any menu is:

(number of fields * 24) + 40

The virtual storage size of the second record of a menu created without extended attributes is:

(menu-line-size * menu-column-size) + (number of fields) + 8

The virtual storage size of the second record of a menu created with extended attributes is:

(menu-line-size * menu-column-size) + (number of fields * 9) + 8

The maximum possible DASD record length of a menu is 65,535 although the usual size is between 1,600 and 5,000 bytes. XMENU menus are kept in variable format (RECFM V) files.

filename **LISTING**

This file contains the printer or LISTING output generated by using the DISK option and printing a copy of the edited menu. The

LISTING file contains variable format records (RECFM V) and ASA carriage control characters.

filename **SCRIPT** This file contains the SCRIPT output generated by using the SCRIPT option and printing a copy of the edited menu. The SCRIPT file is of variable record format (RECFM V), and contains SCRIPT control tags and a "screen print" of the menu.

filename **TEXT** This file contains an object file (similar to the output of the assembler or compilers) generated from the *filename* MENU file. You can load this file as part of your application module and either pass its in-storage addresses to MLOAD, or use it as part of a non-XMENU application. There are four external reference names in the object deck:

filename The name by which the file is loaded. This name points to the first byte of the loaded deck.

*filename***L** Points to a fullword containing the length in bytes of the output data.

*filename***I** Points to the data contained in The first record of a menu file. This address would be passed to MLOAD to load an in-storage menu.

*filename***O** Points to the actual output image, which is contained in the second record of a menu file. This address would be passed to MLOAD to load an in-storage menu.

The *filename* TEXT file is created when using XMEDIT with either the MENTEXT or TEXT XMEDIT options.

Appendix C. Attribute types and values

Field attributes that can be assigned in XMEDIT include the following types and associated values:

Protection Possible values include the following:

- **PROT** To specify a protected field where input is not accepted.
- **UNPROT** To specify an unprotected field where input is accepted.
- **NUMERIC** To specify an unprotected field where only numeric data is accepted. If your terminal does not have the numeric lock feature, a field with this attribute will be treated as **UNPROT**.

Intensity Possible values include the following:

- **BRIGHT** To specify an intensified display field. **BRIGHT** fields are always selector pen detectable.
- **DIM** To specify a normal display field, not detectable with a selector pen.
- **LGHTPEN** To specify a normal display field that is detectable with a selector pen. On terminals equipped with one, the CURSOR SEL key has the same function as the selector pen.
- **DARK** To specify a non-display, or invisible, field (such as one to accept user passwords, for example).

SKIP Possible values include the following (protected fields only):

- **SKIP** To specify that the protected field will be skipped over by the cursor during user input. An unprotected field cannot be set to **SKIP**.
- **NOSKIP** To specify that the protected field will not be skipped over by the cursor during user input.

MDT Possible values include the following:

- **MDT** To set the Modified Data Tag at menu output.
- **NOMDT** To clear the Modified Data Tag at menu output.

If you are creating a menu with **extended attributes**, three additional attribute types are possible:

Color The extended color values include the following:

BLUE
GREEN
RED
PINK
YELLOW
TURQUOISE
WHITE

DEFCOL The hardware default color for the field's protection and intensity is used.

Highlighting The extended highlighting values include the following:

BLINK

UNDERSCORE To underline the entire field. The entire field is underscored, not just the non-blank characters within a field.

REVERSE For reverse video.

DEFHI For no extended highlighting.

Programmed Symbol Set

This attribute type is used to enter a specific logical programmed symbol set value. This value may be zero (00) for the default symbol set, or a hexadecimal number from X'40' to X'EF'. If a non-zero value is used, a symbol set with a matching number must be loaded before the menu is displayed. If not, the default symbol set is used to display the field. If a set with the same number is already loaded into the device, it is displayed with unpredictable results.

Appendix D. Assigning attributes using the OLDWAY full-screen menus

You can assign attributes from Input mode in two ways: with the default attribute action bar and pull-down windows ("Using the attribute action bar and pull-down windows" on page 41), or, if you specified the OLDWAY option, with a full-screen menu. You can switch between the two ways of entering attributes by entering the SET OLDWAY OFF|ON subcommand on the COMMAND pop-up window (see "The XMEDIT command line" on page 50 and "COMMAND subcommand" on page 66 for more information).

If you are using the OLDWAY option, or have SET OLDWAY ON, the Set/Change A Field's Attributes screen is used to enter field attribute values and is displayed when you press PF6/PF18 from Input mode. To assign attributes using the full-screen menus, simply type over any of the highlighted default attributes with your choice of appropriate alternatives. Once you have assigned the attributes you desire, press ENTER to return to Input mode.

The following screen is displayed if you are creating a menu without extended attributes.

```
---------------------- Set/Change A Field's Attributes ----------------------

Menu name  - *XMENU2*   Field name -                Field length - 0

Field offset   85 , Line   2 , Column  6


     *
 To: |

Attributes: Protection  UNPROT     PROT, UNPROT or NUMERIC

            Intensity   DIM        BRIGHT, DIM, DARK or LGHTPEN

            SKIP        NOSKIP     SKIP or NOSKIP

            MDT         NOMDT      MDT or NOMDT

Enter attributes then press ENTER. To leave an attribute as it is press ENTER
without modifying it. To leave all attributes unchanged, press PF03.
```

Figure 29. OLDWAY attribute selection menu

The next screen shows the field definition menu that is displayed when extended attributes are used. Notice the additional selections that are available with extended attributes. Also notice that PF keys are assigned for each extended attribute. You can simply press the PF key associated with the extended attribute you desire and XMEDIT will make the change directly into the appropriate field for you.

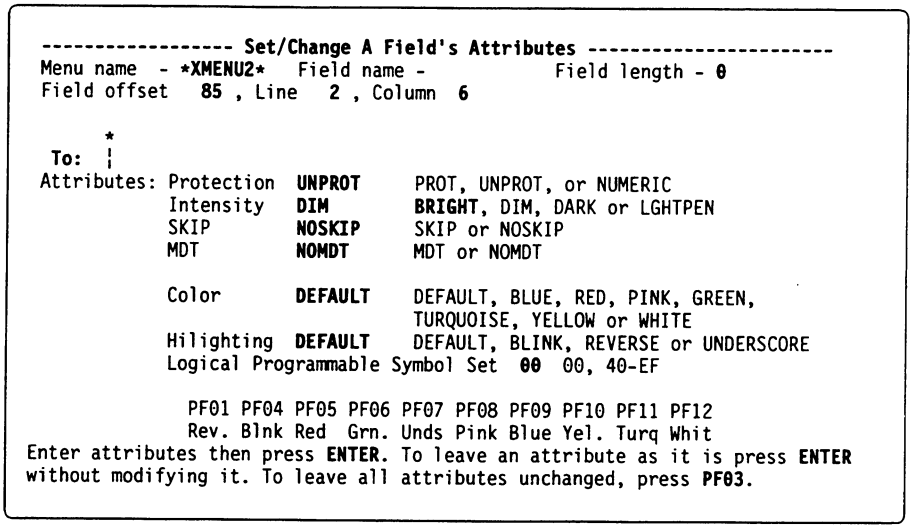


Figure 30. OLDWAY extended attribute selection menu

The following fields are displayed on the full-screen menus and cannot be modified:

Menu name The name of the menu being modified. Although *XMENU2* appears in this field, XMENU "knows" the name of your menu.

Field name The name of the field being modified (if a name has been assigned).

Field length The number of characters in the field.

Field offset The line and column position where the field begins.

You can also review or modify attributes from the full-screen Set/Change a Field's Name screen.

Appendix E. Naming fields by using the OLDWAY full-screen menus

You can name fields from Input mode in two ways: with the default pop-up window ("Name a field" on page 45), or, if you specified the OLDWAY option, with a full-screen menu. You can switch between the two ways of naming fields by entering the SET OLDWAY OFF|ON subcommand on the COMMAND pop-up window (see "The XMEDIT command line" on page 50 and "COMMAND subcommand" on page 66 for more information).

To name a field using the full-screen Set/Change a Field's Name menu, you follow these steps:

1. Place the cursor over the field marker character—the split vertical bar—that marks the beginning of the field you want to name.
2. Press PF12/PF24 to display the Set/Change a Field's Name screen (see the following figure).
3. Type a valid field name in the space provided.
4. Press ENTER.

```
------------------------ Set/Change a Field's Name --------------------------

Menu name  - *XMENU2*

Field name - ▌        - Enter a 1-7 character unique field name.


Field offset   85 , Line   2 , Column  6   Field length -     9


*
To:  ¦         ¦

Attributes: Protection  UNPROT   PROT, UNPROT or NUMERIC
            Intensity   DIM      BRIGHT, DARK or LGHTPEN
            SKIP        NOSKIP   SKIP or NOSKIP
            MDT         NOMDT    MDT or NOMDT

Place/change field name then press ENTER. To remove a name from a field press
ERASE EOF then ENTER. To exit, press PF03.
```

Figure 31. The Set/Change a Field's Name Screen

Notice that you can review and modify the field's attributes from this screen as well. If you created the menu using extended attributes, this full-screen menu would include those attributes, just like the Set/Change A Field's Attributes screen in Figure 30 on page 124.

After you press ENTER from the field naming screen, the split vertical bar that appeared at the beginning of the input field has been replaced by a tilde (~), indicating that the field has a name:

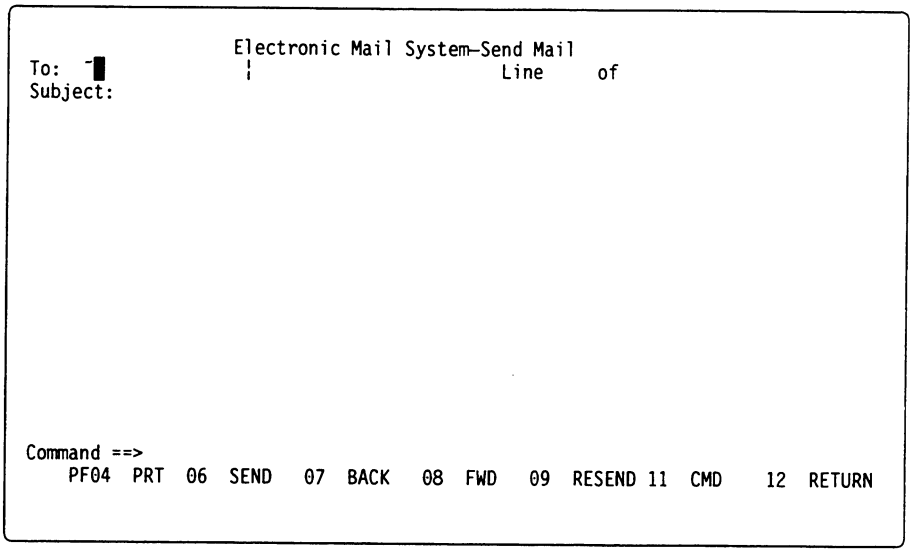


Figure 32. Sample menu showing a named field

Appendix F. Changing the defaults by editing PROFILE XMEDIT

If you want to change XMEDIT's default terminal key functions, you can do so by editing the PROFILE XMEDIT file.

PROFILE XMEDIT contains a list of all the current PF key settings. This file typically resides on the same minidisk as the XMENU product files, so you have it on an accessed disk. Although it is unlikely to be necessary, you can edit this file by entering the command `XEDIT PROFILE XMEDIT`.

Note: With XMENU Version 2 Release 2, the name and format of the editor profile changed from XMENU $PROFILE to PROFILE XMEDIT to provide compatibility with the new XMEDIT macro facility. Any users having their own XMENU $PROFILE file need to convert it to a PROFILE XMEDIT file. A REXX utility, CVTXPROF, has been provided for this purpose. This utility reads an existing XMENU $PROFILE and converts it to a new PROFILE XMEDIT file—the old file is preserved. Because the XMEDIT editor doesn't support the old file, if users do not convert it, their profiles will never be executed.

A sample PROFILE XMEDIT file follows:

```
&TRACE OFF
*
*  XMEDIT Default Profile File
*  Copyright 1989, Relay Technology. All rights reserved.
*
*    Command Formats are:
*        SET CHAR char1 char2
*        SET EXT ON
*        SET ALT ON
*        SET SCREEN INPUT key command <Attribute definitions>
*        Keys are: PF01-PF24, PA1-PA3, ENTER, TESTREQ
*        Commands are:
*           DISPLAY SUBSET HELP QUIT SAVE FILE HELPMENU
*           UPCASE LOWCASE CURSOR NAME DESTROY PRINT EDIT
*           COMMAND
*           FWDSPACE nn BACKSPACE nn
*           UP nn DOWN nn LEFT nn RIGHT nn
*           FIELD <attributes>
*        Attributes are:
*           PROT UNPROT NUMERIC BRIGHT DIM DARK LGHTPEN
*           SKIP NOSKIP MDT NOMDT
*           BLUE GREEN RED PINK YELLOW TURQUOISE WHITE
*           BLINK UNDERSCORE REVERSE
*           PS 00 or PS 40-EF
*        Undefined attributes default to:
*           UNPROT DIM NOSKIP NOMDT DEFCOL DEFHI PS 00
*
* SET CHAR 6A A1
* SET EXT ON
* SET ALT ON
*   PA Keys, Test Request, Enter...
SET SCREEN INPUT PA1 DISPLAY
SET SCREEN INPUT PA2 DESTROY
SET SCREEN INPUT PA3
SET SCREEN INPUT TESTREQ DISPLAY
SET SCREEN INPUT ENTER EDIT
*  PFK 01-12...
SET SCREEN INPUT PF01 COMMAND
SET SCREEN INPUT PF02 QUIT
SET SCREEN INPUT PF03 FILE
SET SCREEN INPUT PF04 FIELD Prot Bright
SET SCREEN INPUT PF05 FIELD Prot Dim
SET SCREEN INPUT PF06 FIELD
SET SCREEN INPUT PF07 FIELD Unprot Bright
SET SCREEN INPUT PF08 FIELD Unprot Dim
* Remove asterisks below if you want PFkeys for scrolling
* menus bigger than your terminal size.
* SET SCREEN INPUT PF07 Up 10
* SET SCREEN INPUT PF08 Down 10
SET SCREEN INPUT PF09 FIELD Unprot Dark
SET SCREEN INPUT PF10 UPCASE
SET SCREEN INPUT PF11 IC
* Remove asterisks below if you want PFkeys for scrolling
* menus bigger than your terminal size.
* SET SCREEN INPUT PF10 Left 10
* SET SCREEN INPUT PF11 Right 10
SET SCREEN INPUT PF12 NAME
*  PFK 13-24
SET SCREEN INPUT PF13 QUERY SCREEN INPUT
SET SCREEN INPUT PF14 QUIT
SET SCREEN INPUT PF15 FILE
SET SCREEN INPUT PF16 FIELD Prot Bright
SET SCREEN INPUT PF17 FIELD Prot Dim
SET SCREEN INPUT PF18 FIELD
SET SCREEN INPUT PF19 FIELD Unprot Bright
SET SCREEN INPUT PF20 FIELD Unprot Dim
SET SCREEN INPUT PF21 FIELD Unprot Dark
SET SCREEN INPUT PF22 UPCASE
SET SCREEN INPUT PF23 IC
SET SCREEN INPUT PF24 NAME
```

To change the setting of a PF or PA key, change the text after the key you want to alter. Any of the XMEDIT subcommands, described in Part 2, The XMENU editor utilities reference, "XMEDIT subcommands" on page 62, can be assigned in PROFILE XMEDIT. These include, but are not limited to, the following frequently-used subcommands:

BACKSPACE *nn* Move the cursor backward *nn* positions toward the top left corner of the menu.

COMMAND Display a pop-up menu allowing you to enter XMEDIT, CMS, or CP commands.

CURSOR Set the initial menu cursor position to the current cursor position.

DESTROY Remove the field at the cursor position.

DISPLAY Enter Display mode, showing how the menu will look when displayed by an application.

DOWN *nn* Scroll the Input and Edit mode screens *nn* lines downward.

EDIT Enter Edit mode.

FIELD *attr* Modify or create fields with specific attribute definitions at the cursor position. These attributes can be: PROT, UNPROT, NUMERIC, BRIGHT, DIM, DARK, LGHTPEN, SKIP, NOSKIP, MDT, NOMDT, DEFCOL, BLUE, RED, PINK, GREEN, TURQUOISE, YELLOW, WHITE, DEFHI, BLINK, REVERSE, UNDERSCORE, or PS *xx*, where *xx* may be a hexadecimal value of X'00' or from X'40' to X'EF'. The default values are UNPROT, DIM, NOSKIP, NOMDT, DEFCOL, DEFHI, and PS 00.

FILE Save the current menu and exit XMEDIT.

FWDSPACE *nn* Move the cursor forward *nn* positions toward the bottom right corner of the menu.

HELPMENU Display a screen that defines the default Input mode PF key definitions.

LEFT *nn* Scroll the Input and Display mode screens *nn* columns left.

LOWCASE Convert data at the cursor position to lowercase.

NAME Name the field at the cursor position or, if not positioned on a field marker, move the cursor forward to the next field marker.

PRINT Print the menu and the field definitions.

QQUIT Exit XMEDIT without saving changes to the menu.

RIGHT *nn* Scroll the Input and Display mode screens *nn* columns right.

SAVE Save the current menu and return to Input mode.

UP *nn* Scroll the Input and Edit mode screens *nn* lines upward.

UPCASE Convert data at the cursor position to uppercase.

Note: You can also specify CHAR to set the characters used to mark fields, EXT to use extended attributes, and ALT to use Alternate Input mode. These are parameters of the XMEDIT SET subcommand. For more information on this subcommand see "SET subcommand" on page 97.

When you finish making changes, file PROFILE XMEDIT on your A-disk by entering the XEDIT subcommand `SET FMODE A` before you enter the `FILE` subcommand. Because you have not changed this file's name, there is no need to change the `Use Profile` field on the XMEDIT options screen. The copy you have just saved on your A-disk will be used each time you enter XMEDIT.

Appendix G. Some useful XMEDIT Macros

FLDLST macro

The FLDLST macro lists all fields defined in a menu. From the FLDLST panel, any and all of a field's attributes can be altered. Changes can be propagated to other fields easily. The field's position and a small portion of the data is displayed for your information.

The FLDLST menu is presented below:

```
Name     Prot   Intens  MDT   Skip    Color   Highlight  PSS Line Col  Data

         UNPROT BRIGHT  NOMDT NOSKIP WHITE   DEFAULT    00  1    2     Help
         PROT   DIM     NOMDT NOSKIP BLUE    DEFAULT    00  1    8
         PROT   DIM     NOMDT NOSKIP BLUE    DEFAULT    00  2    1     ________
S1       UNPROT DIM     NOMDT NOSKIP TURQUOI UNDERSCORE 00  3    1
S1C1     PROT   DIM     NOMDT NOSKIP WHITE   DEFAULT    00  3    3     1. Selecti
ERRMSG   PROT   BRIGHT  NOMDT NOSKIP RED     DEFAULT    00  18   1
         UNPROT DARK    NOMDT NOSKIP DEFAULT DEFAULT    00  18   2
CMDPREF  PROT   DIM     NOMDT NOSKIP GREEN   DEFAULT    00  19   1     Command ==
CMDLINE  UNPROT DIM     MDT   NOSKIP TURQUOI UNDERSCORE 00  19   14
PFKLINE  PROT   DIM     NOMDT NOSKIP BLUE    DEFAULT    00  20   1




PF: 03=Exit   04=Back-1/2   05=Fwd-1/2   07=Back   08=Fwd   09=Data
```

Figure 33. FLDLST macro display

Usage

This macro is used to list all fields within a menu. It can be executed from the XMEDIT command line, or invoked with a PF key. To assign a key, place the following command in your PROFILE XMEDIT or issue it from the command line:

```
SET SCREEN INPUT PFnn FLDLST
```

For information on the actual FIELD settings, issue:

```
HELP XMEDIT FIELD
```

Field names are changed or added by simply typing the name in the appropriate column.

Attribute values can be changed by simply typing the new value in place of the old one. FLDLST recognizes the equal sign ('=') as an indication to repeat this change on successive fields. For example:

```
Name     Prot    Intens  MDT   Skip    Color  Highlight  PSS Line Col  Data

S1       UNPROT DIM      NOMDT NOSKIP TURQUOI UNDERSCORE 00  3    1
S1C1     PROT   DIM      NOMDT NOSKIP WHITE   reverse    00  3    3    1. Selecti
ERRMSG   PROT   BRIGHT   NOMDT NOSKIP RED     DEFAULT    00  18   1
         UNPROT DARK     NOMDT NOSKIP DEFAULT DEFAULT    00  18   2
CMDPREF  PROT   DIM      NOMDT NOSKIP GREEN   =EFAULT    00  19   1    Command ==
CMDLINE  UNPROT DIM      MDT   NOSKIP TURQUOI UNDERSCORE 00  19   14
PFKLINE  PROT   DIM      NOMDT NOSKIP BLUE    =EFAULT    00  20   1


PF: 03=Exit   04=Back-1/2   05=Fwd-1/2   07=Back   08=Fwd   09=Data
```

In the above example, field ***S1C1*** is changed from default extended highlighting to reverse-video. Because '=' appears on fields ***CMDPREF*** and ***PFKLINE***, they too will be changed to reverse-video.

PF Key commands for FLDLST is described below:

PF Key	**Description**
PF03 (Exit)	Returns to the XMEDIT session.
PF04 (Back-1/2)	Pages back one half of a page.
PF04 (Fwd-1/2)	Pages forward one half of a page.
PF07 (Back)	Pages back a full page.
PF08 (Fwd)	Pages forward a full page.
PF09 (Data/List)	Switches between normal list and expanded data panels. The expanded data panel shows the first 78 bytes of the field's data.

FLDSET macro

The FLDSET macro creates or updates a menu field. From the FLDSET panel, any and all of a field's attributes can be set or altered. Existing field definitions can, if desired, be duplicated and placed in other locations.

Note: If you copy a field that has a name assigned, the name will not be replicated, since XMENU allows only unique field names. You can, however, type in a newname before pressing ENTER.

The FLDSET menu is presented below:

```
Field Attribute Display - Lines:     Columns:     Bytes:
Name                      Field's name
Protection                PROT, UNPROT, NUMERIC
Intensity                 BRIGHT, DIM, DARK, LGHTPEN
Skip                      SKIP, NOSKIP
MDT                       MDT, NOMDT
ExtColor                  BLUE  GREEN  RED  YELLOW  PINK  TURQ  WHITE
ExtHighlight              BLINK  UNDER  REV NOHI
PSS                       Program symbol set (nn)

Skip      fields to next
Line:       Column:       Position:
Field Command:
PF:  2=Goto   3=Quit  4=Move  7=Prev   8=Next  10=Destroy  Enter=Update
```

Figure 34. FLDSET macro display

Usage

This macro is used to display, alter, delete, and copy fields within a menu. It can be executed from the XMEDIT command line, or invoked with a PF key. To assign a key, place the following command in your PROFILE XMEDIT or issue it from the command line:

```
SET SCREEN INPUT PFnn FLDSET
```

For information on the actual FIELD settings, issue:

```
HELP XMEDIT FIELD
```

User input and PF Key commands for FLDSET is described below:

Field/PF Key	Description
Skip	Allows skipping 'nn' fields after taking the designated action. This is useful if you are making changes to fields in one column, and wish to skip over a fixed number of fields to get to the next column occurrence.
Line	Specifies the line (along with column specification) to which the current field is to be copied or, if used with the 'Goto' function, reposition the cursor only.

Column	Specifies the column (along with line specification) to which the current field is to be copied or, if used with the 'Goto' function, reposition the cursor only.
Position	Specifies the offset relative to the beginning of the menu to which the current field is to be copied or, if used with the 'Goto' function, reposition the cursor only.
PF02 (Goto)	Repositions the cursor to the specified line/column or position value without making field changes. If a field exists at that location, its value is displayed.
PF03 (Quit)	Returns to the XMEDIT session.
PF04 (Move)	Moves this field to the location specified by line/column or position. If a field exists at the new location, it is replaced.
PF07 (Prev)	Backs up to the previous field.
PF08 (Next)	Advances to the next field.
PF10 (Destroy)	Deletes the field at the current location.
Enter (Update)	Performs any field changes and remains at the current location.

FLDCHG macro

The FLDCHG macro allows global changes to fields in a menu.

The FLDCHG menu is presented below:

```
                          Global Attribute Change

            Change this ...  to this

Protection                =>          ( PROT UNPROT NUMERIC )
Intensity                 =>          ( BRIGHT DIM DARK LGHTPEN )
Skip                      =>          ( SKIP NOSKIP )
MDT                       =>          ( MDT NOMDT )
ExtColor                  =>          ( BLUE GREEN RED YELLOW PINK TURQ WHITE )
ExtHighlight              =>          ( BLINK UNDER REV NOHI )
PSS                       =>          ( 00, 40-EF )
            (* = anything)

Limit changes to        occurrences
Start at current cursor position?     (line      column     )
                                   (N = scan all fields)


                     Press Enter to update, PF03 to Quit
```

Figure 35. FLDCHG

Usage

This macro is used to effect specific changes to all fields on a menu. It can be executed from the XMEDIT command line, or invoked with a PF key. To assign a key, place the following command in your PROFILE XMEDIT or issue it from the command line:

```
SET SCREEN INPUT PFnn FLDCHG
```

The "Change this..." column allows specification of attributes to be used to identify eligible fields that will be changed. For information on the actual FIELD settings, issue:

```
HELP XMEDIT FIELD
```

User input and PF Key commands for FLDCHG is described below:

Field/PF Key	Description
'*' (asterisk)	Specifies that any value in the 'Change this' column qualifies for the change to take effect. For example, if you wanted to change all PROTECTED NOSKIP fields to BLUE regardless of their current color, place an asterisk in the left ExtColor column, PROT in the left Protection column, NOSKIP in the left Skip column, and BLUE in the right column. The attributes specified in the left column are additive, i.e., the field must meet all criteria to be eligible for change.
Limit changes to .. occurrences	Specifies the global change is limited to the number of fields specified.
Start at current cursor position?	Specifies the changes should start from where the cursor is positioned on the menu. Useful in combination with change limit.
PF03 (Quit)	Returns to the XMEDIT session without making designated changes.
ENTER	Effects the specified changes.

GENCOLS macro

The GENCOLS macro generates columns of fields within a menu.

The GENCOLS menu is presented below:

```
                        Enter column information below:

Number of rows:
Starting line number:
Starting var number:

Column     Column    Column
name       length    attributes

.....      ...       .........
.....      ...       .........
.....      ...       .........



PF:  03=Exit
```

Figure 36. GENCOLS

Usage

This macro is used to generate columns of fields. Names generated are in REXX stem format, eg. *NAME.1*. It can be executed from the XMEDIT command line, or invoked with a PF key. To assign a key, place the following command in your PROFILE XMEDIT or issue it from the command line:

```
SET SCREEN INPUT PFnn GENCOLS
```

For information on attribute settings, issue:

```
HELP XMEDIT FIELD
```

User input and PF Key commands for GENCOLS is described below:

Field/PF Key	Description
Number of rows:	Specifies the number of rows (lines) to generate.
Starting line number:	Specifies the line upon which to start the columns.
Starting var number	Specifies the first number to use for the REXX variable name.
Column name	Specifies the prefix to use for the REXX variable name
Column length	Specifies the input length of this field. GENCOLS computes the line length to ensure the specification will fit.

Column attributes	Specifies the field attributes for this field. GENCOLS recognizes the use of '=' to propagate the attributes to successive fields.
ENTER	Generates the columns specified
PF03 (Exit)	Exits without generating the columns.

The following GENCOLS invocation was used to create the menu used by FLDLST. Refer to the FLDLST macro to examine the results.

```
                    Enter column information below:

Number of rows: 20
Starting line number: 3
Starting var number: 1

Column    Column   Column
name      length   attributes

name      7        unprot white
prot      6        =
ints      7        =
mdt       5        =
skip      6        =
colr      7        =
high      10       =
pss       3        =
line      4        =
col       4        =
data      11       =


PF:  03=Exit
```

Figure 37. GENCOLS invocation used to create FLDLST MENU

GENPAIR macro

The GENPAIR macro generates pairs of fields within a menu.

The GENPAIR menu is presented below:

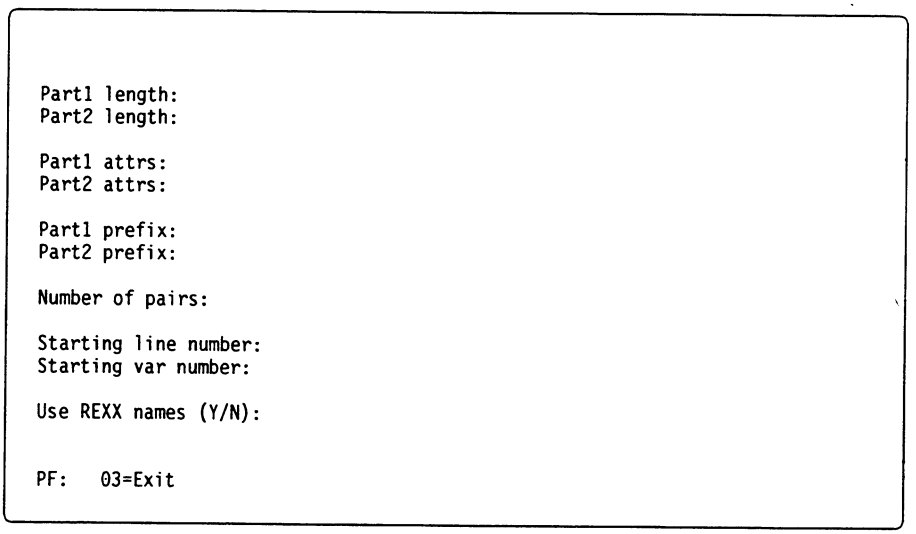


Figure 38. GENPAIR

Usage

This macro is used to generate pairs of fields. These pairs are most often used as corresponding input/output fields, such as a keyword/value pairs. It can be executed from the XMEDIT command line, or invoked with a PF key. To assign a key, place the following command in your PROFILE XMEDIT or issue it from the command line:

```
SET SCREEN INPUT PFnn GENPAIR
```

For information on attribute settings, issue:

```
HELP XMEDIT FIELD
```

User input and PF Key commands for GENPAIR is described below:

Field/PF Key	Description
Part1 length:	Specifies the length of the first field in the pair.
Part2 length:	Specifies the length of the second field in the pair.
Part1 attrs:	Specifies the attributes for the first field in the pair.
Part2 length:	Specifies the attributes for the second field in the pair.
Part1 prefix:	Specifies the name prefix for the first field in the pair.

Part2 prefix:	Specifies the name prefix for the second field in the pair.
Number of pairs:	Specifies the number of pairs to be generated. GENPAIR will calculate how many pairs of fields will fit on a line based on the menu's size.
Starting line number:	Specifies the first line to be used to contain the field pairs.
Starting var number:	Specifies the first number to use for the field names.
Use REXX names (Y/N):	Specifies whether the variable number is separated from the name prefix by a period to form REXX stems.
ENTER	Generates the columns specified
PF03 (Exit)	Exits without generating the columns.

The following GENPAIR invocation will generate 30 pairs of fields named IN.1 - IN.30 and OUT.1 - OUT.30 starting on line 4 of the menu:

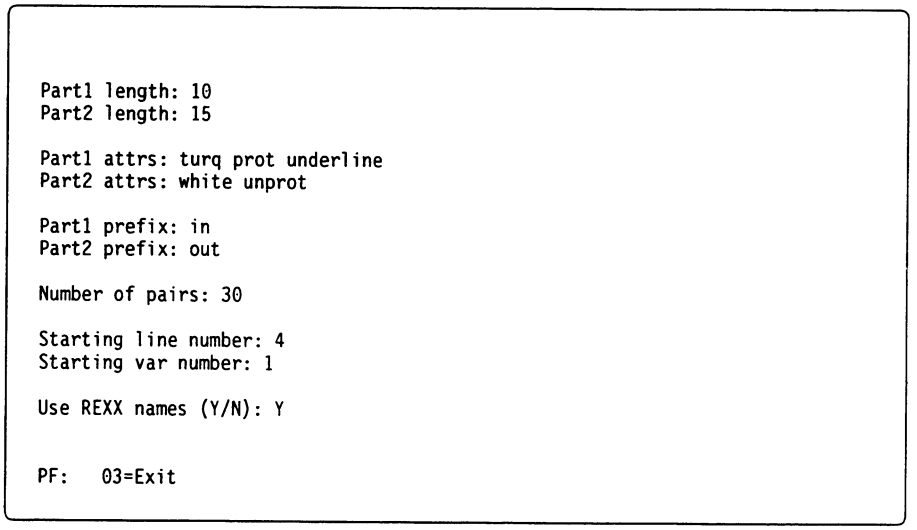


Figure 39. GENPAIR

Index

Special Characters

Numerics

A

B

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y